For Reference

NOT TO BE TAKEN FROM THIS ROOM

705-142

Ex libris
universitatis
albertaeasis

QUAECUMQUE VERA

THE UNIVERSITY OF ALBERTA

SOME OPTIMUM CRITERIA FOR INFORMATION RETRIEVAL

by

Ⓒ WILFRED C.C. KO

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL, 1970

Thesis
1975F
149

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled SOME OPTIMUM CRITERIA FOR INFORMATION RETRIEVAL, submitted by Wilfred C.C. Ko in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

The need for effective methods in document searches is first examined, and then four optimum criteria are presented. Optimization is accomplished by finding a set of optimum weights for the keywords in the question. Each of the four optimum procedures is discussed in some detail. Examples are also presented for illustration and comparison. The least square criterion seems to be a promising optimum procedure.

ACKNOWLEDGEMENTS

I wish to express my appreciation to Professor H.S. Heaps for his advice and guidance given me at all stages of this study. I am also indebted to Dr. I.N. Chen for suggesting an extended matched search leading to the cross correlated version. Thanks are also due to the National Research Council for providing financial assistance to enable me to undertake this research.

TABLE OF CONTENTS

Page

CHAPTER 1 INTRODUCTION 1

1.1 Need for Optimization 1

1.2 Analogy in Filter Theory 3

1.3 A Simple Model 4

1.4 Form of Question and Data Base 6

CHAPTER 2 MATCHED SEARCH 8

2.1 Criterion of Maximum Average Response 8

2.2 Power-Series Approximation 11

2.3 Flow Chart of Matched Search 14

2.4 Application of Matched Search 17

2.5 Effect of Addition Cross-Correlated Terms 21

CHAPTER 3 LEAST SQUARE SEARCH 28

3.1 The Search Criterion 28

3.2 The Search Procedure 29

3.3 The Question Form 32

3.4 An Example 34

CHAPTER 4 TWO-PARAMETER SEARCH 37

4.1 The Search Criterion 37

4.2 Optimization Procedure 38

4.3 Question Formulation 42

4.4 An Example 43

Page

CHAPTER 5 COMPARISON AND CONCLUSION 46

5.1 Comparative Study of Search Procedures 46

5.2 Discussion and Conclusion 66

REFERENCES 71

APPENDIX A 74

APPENDIX B 95

APPENDIX C 98

APPENDIX D 99

LIST OF TABLES

Table		Page
3.1	Result of least square search	36
4.1	Result of 2-parameter search	44
5.1	Matched Search A	49
5.2	Cross-correlated matched search A	50
5.3	Least square search A	51
5.4	2-parameter search A	52
5.5	Matched search B	54
5.6	Cross-correlated search B	55
5.7	Least square search B	56
5.8	2-parameter search B	57
5.9	Matched search C	59
5.10	Cross-correlated search C	60
5.11	Least square search C	61
5.12	2-parameter search C	62
5.13	Matched search D	63
5.14	Cross-correlated search D	64
5.15	Least square search D	65
5.16	2-parameter search D	66

LIST OF FIGURES

Figure		Page
1.1	Document processor in a simple system	5
1.2	Automatic retrieval system with optimization	6
2.1	Flow Chart of matched search	16

CHAPTER 1

INTRODUCTION

1.1 Need for Optimization

Information retrieval is a field concerned with the problems of organization and storage, and file structure, as well as retrieval techniques. With the advent of computers a considerable amount of attention has been focused on the development of information retrieval systems in which the computer plays a basic role. The computer not only permits mechanized handling of data by existing methods, but it also opens up paths for new approaches which are otherwise not possible. In this respect, a few experimental interactive systems have been developed with varying degrees of success [1 to 4].

The majority of interactive systems operate in a real time, man-machine dialogue, mode in which the system accepts requests either in precoded form or in natural language. In the analysis phase the request is analyzed and correlated to documents in storage. Highly correlated documents are produced as hits. In some cases, such as in the SMART system [4], the user may be able to control the correlation measures by

specifying weights in the request. Depending on whether the user is satisfied with the system's initial response, he may reformulate the request in order to obtain an improved response. To do this some way of evaluating system performance is needed.

The system may be considered to be efficient if the response produces highly useful items in terms of relevance and quantity. It is thus possible to define two ratios "Recall" and "Precision" for the evaluation of a system's performance. Recall is the proportion of relevant items in storage that are successfully retrieved, while Precision is the proportion of retrieved documents that are actually relevant.

Although high Recall and Precision values may be chosen as the measure of a good response, it requires a manual examination of the collection in storage to determine the total number of relevant items for each given request. While this method of evaluation may not cause difficulties in small experimental systems, it is completely unsatisfactory in normal environments where very large data collections are likely to occur. To overcome this shortcoming Salton [5] has suggested sampling techniques, or the use of approximate Recall values based upon search requests on limited source documents in the collection. In spite of his suggestions

it is felt that Recall and Precision measurements will be regarded as idealized theoretical concepts, and are not useful tools in practice.

A more direct approach to the prediction of system efficiency is to find a set of optimum weights for each request. Through this optimization it is intended to restrict the system response to the most relevant items for that request.

Barkle [6] has discussed a method of arriving at the optimum weights in terms of a probability ratio, called the specificity, which relates the frequency of occurrence of a keyword in relevant items to the frequency of occurrence of it in the whole collection to be searched. The weights are then made proportional to the specificities. It is expected that the higher the specificity, the greater is the probability that an item containing the keyword will be relevant.

1.2 Analogy in filter theory

An alternative approach to the problem of obtaining optimum weights may be made through the use of filtering techniques. Suppose a signal f(t) is received and it contains a message g(t). It is required that g(t) should be extracted from f(t) with as little distortion as possible. One method is to pass f(t)

through a filter designed to minimize the root-mean-square error of the noise which is $f(t)-g(t)$. This results in the well known Wiener filter [7,8]. An alternative method is to maximize the ratio between the peak amplitude of the signal and the root-mean-square of the noise [9], this will be referred to as the matched procedure. In either case, realizable filters can be constructed to suppress the noise background as far as possible so that the desired message can be detected.

In like manner, an information retrieval system can be designed with the aim of giving the best possible response to a request. In the present study the corresponding theory of filters will be examined very briefly and extended to lead to the design of an optimal retrieval system. The optimum response can then be defined as the best subset of documents retrieved with minimum distortion, in the sense that the Recall and Precision values are close to unity.

1.3 A simple model

In its simplest form, a retrieval system may be considered to include a document processor as shown in Fig. 1.1. The document processor can be regarded as a device with parameters $\underline{W}$ as determined by each input

request. Output response will be obtained after each document $\underline{Z}(n)$ has been processed.

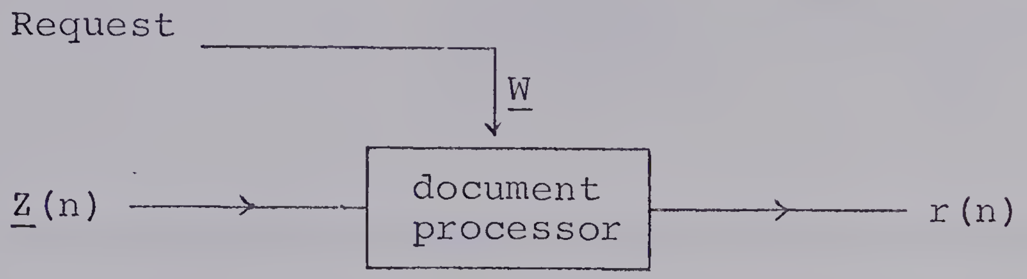

Fig. 1.1 Document processor in a simple system

Suppose a request for information is converted to a specification of process parameters $\underline{W}$ which are then used to process each document $\underline{Z}(n)$, the choice of $\underline{W}$ is to be made automatically according to some optimizing procedure as shown in Fig. 1.2, and the processing of each document is also to be made automatic. Thus, given a retrieval criterion, each request determines an optimum set of weights $\underline{W}$ which are used to process each document in the entire collection. It is evident that different requests determine different sets of weights.

A threshold value may be chosen so that the response of each document may be compared with it and a ratio t obtained. Each value of t is considered to be the system's estimate on the relevancy of the document to the request.

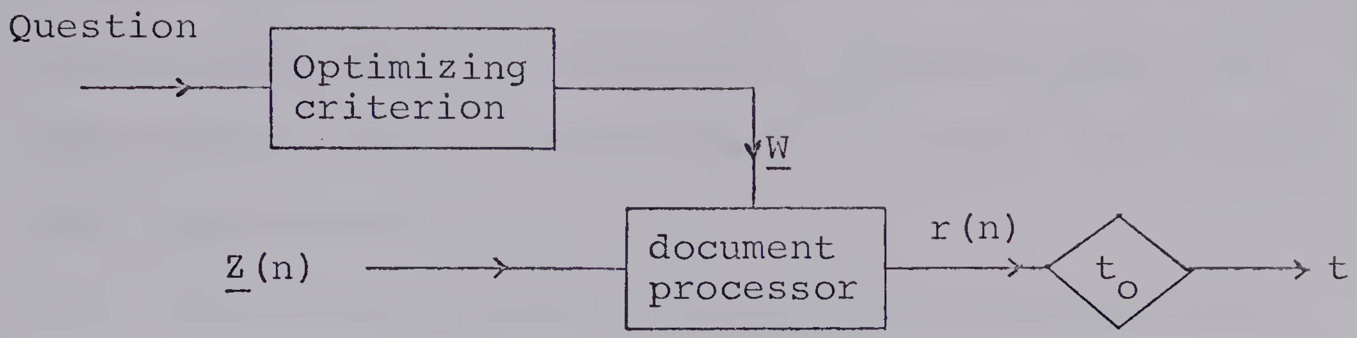

Fig. 1.2 Automatic retrieval system with optimization

1.4 Form of question and data base

The user's request may be in one of several forms. It may specify a list of vectors $\underline{x}$ which represent actual titles obtained from a previous search, or partial search, and which are found relevant to the question. Alternatively the vectors may represent highly relevant keywords chosen for the question. A mixture of these two forms of request may also form a valid question.

It will be assumed that the question consists of keywords coded in truncated form, the use of truncation having proved to be useful [9]. In this form, the question can be considered as a vector, the components of which are the weights that may be negative, positive or zero. The simplest case is when they take on binary values. Then with reference to a chosen set of parameters the i^{th} element will be 1 if a keyword in the question matches the i^{th} parameter in the given set and

it will be zero if the keyword does not match. Documents may be represented by vectors in a similar manner, and such representation proves to be very useful for mathematical operations.

The data base used is the ACOUS tape prepared by the Department of Computing Science at the University of Alberta. It is made up of approximately 7,000 entries, each of which refers to publications that appeared in one of the following Journals during the period 1954-1969: *Journal of Acoustic Society of America*, *Acoustica*, *Soviet Physics (Acoustic)*, and *Journal of Fluid Mechanics.* All the articles are in the fields of acoustics, fluid mechanics, or related areas. It is believed that this data base is large enough to produce meaningful results for the present investigation. The choice of a rather specialized area instead of the whole of the scientific literature permits the results to be analyzed unambiguously and efficiently. An example of the form of the references on the ACOUS tape is shown by the following list:

```
SOV 57 0003 0017ALMUK PULSE METHO MEASU NONUN MOTIO MAGNE
TAPE
ACO 57 0029 0654LYON  LOW   FREQU RADIA LOAD BASS REFLE
SPEAK
ACO 60 0032 0709ZANKE DIFFR LIGHT TWO   ULTRA WAVES SAME
FREQ
ACU 59 0009 0403CONNO MODEL AID   ACOUS DESIG AUDIT
```

The first three letters indicate the journal name, followed by the year. The next four digits give the volume number which is followed by the page number. Immediately following the numerals are author name(s) and title words. These are truncated to a maximum of five characters. Although truncated keywords may be ambiguously interpreted in some cases, no difficulties have been encountered in the interpretation of results obtained in searches on the ACOUS tape.

CHAPTER 2

MATCHED SEARCH

2.1 Criterion of maximum average response

The signal detection technique employing matched linear filters will be used in the retrieval problem presented here. The search for relevant items will be treated similarly to the detection of a signal within a noise background. It will be assumed that the documents are in sufficiently large quantity to make the analogy plausible.

As mentioned previously, the document processor is linear so that the response to the n^{th} document is

$$r(n) = \sum_i w_i \, z_i(n) \tag{2.1}$$

where the w_i and $z_i(n)$ respectively denote components of the vectors $\underline{W}$ and $\underline{Z}(n)$ of Fig. 1.1. Thus the w_i describe the request, while the $z_i(n)$ describe the n-th document.

The retrieval criterion is that r(n) be as large as possible for documents relevant to the request. The measure of largeness is in reference to the average response to the documents of the whole set.

The mean square response of the document processor to the whole set of N documents is

$$r_m^2 = \frac{1}{N} \sum_n r^2(n)$$

$$= \sum_i \sum_j c_{ij} \, w_i \, w_j \qquad (2.2)$$

where

$$c_{ij} = c_{ji} = \frac{1}{N} \sum_n z_i(n) \; z_j(n) \quad . \qquad (2.3)$$

The c_{ij} may be regarded as the cross-correlation measure between the i-th and j-th keywords in each document.

Without loss of generality, assume the request consists of M vectors $\underline{x}$ representing a set of actual, or hypothetical, documents. The mean response to $\underline{x}$ is

$$r_x = \frac{1}{M} \sum_i w_i \sum_m x_i(m)$$

$$= \sum_i w_i \, \bar{x}_i \qquad (2.4)$$

where

$$\bar{x}_i = \frac{1}{M} \sum_m x_i(m) \quad . \qquad (2.5)$$

Hence the vector $\underline{\bar{X}}$ denotes the average of the M input titles.

The set of weights $\underline{W}$ will be considered as chosen optimally if the response r_x^2 is a maximum in comparison to r_m^2. Thus for each weight w_i, the following equation must hold,

$$\frac{\partial}{\partial w_i}\left[\frac{(\sum_i w_i \bar{x}_i)^2}{\sum_i \sum_j c_{ij} w_i w_j}\right] = 0 \text{ , for each } i \tag{2.6}$$

or $$\bar{x}_i - \lambda \sum_j c_{ij} w_j = 0 \tag{2.7}$$ *

where λ denotes the stationary value of r_x^2/r_m^2.

In matrix notation equations (2.2), (2.4) and (2.7) can be expressed as

$$r_m^2 = \underline{W}_T \, C \, \underline{W} \tag{2.8}$$

$$r_x = \underline{W}_T \, \underline{\bar{X}} \tag{2.9}$$

and $$C \, \underline{W} = \frac{1}{\lambda} \, \underline{\bar{X}} \quad . \tag{2.10}$$

The stationary value of $(r_x/r_m)^2$ is then

$$\frac{(W_T \, \underline{\bar{X}})^2}{W_T \, C \, W} = \underline{\bar{X}}_T \, C^{-1} \, \underline{\bar{X}} \tag{2.11}$$

where the subscript T indicates a transpose and the superscript -1 denotes an inverse operation.

With the set of w_i chosen to satisfy equation (2.10) the processor is optimum in the sense that for the given set of vectors $\underline{x}$, it gives the maximum average response r_x with respect to the root mean square response r_m. Apart from a constant of proportionality, which

---

* Derivation of equation (2.7) appears in Appendix C.

does not affect the value of r_x/r_m, the set of optimum weights is obtained as

$$\underline{W} = C^{-1} \underline{\bar{X}} . \tag{2.12}$$

Thus the processor response to any document Z(n) is given by

$$r = \underline{W}_T \underline{Z} = \underline{\bar{X}}_T C^{-1} \underline{Z} . \tag{2.13}$$

With the value of the threshold level chosen as the root-mean-square response r_m, the relevance measure t may be determined as

$$t = \frac{r}{r_m} . \tag{2.14}$$

A cut-off value t_o may be defined as

$$t_o = \frac{r_x}{r_m} \tag{2.15}$$

so that for each document processed, a comparison may be made between the values t and t_o. A document with a value of t considerably smaller than t_o will be eliminated as being non-relevant to the request.

2.2 Power-series approximation

It appears that in equation (2.12) document processing requires a matrix inversion to determine C^{-1},

a process in which computation may become excessively cumbersome when large matrices are encountered. It is possible, however, to estimate C^{-1} by an approximation that does not require a formal matrix inversion. Let

$$C = D + N \tag{2.16}$$

where D is a diagonal matrix with elements

$$d_{ii} = c_{ii} \tag{2.17}$$

and N is a square matrix whose elements are

$$n_{ij} = \begin{cases} c_{ij} , & i \neq j \\ 0 & i = j \end{cases} . \tag{2.18}$$

Equation (2.13) may be expressed in terms of D and N,

$$\begin{aligned} r &= \underline{\bar{X}}_T \, C^{-1} \, \underline{Z} \\ &= \underline{\bar{X}}_T \, (I + D^{-1} N)^{-1} D^{-1} \underline{Z} \end{aligned} \tag{2.19}$$

where I is the identity matrix.

In terms of a power series the last equation becomes

$$\begin{aligned} r = {} & \underline{\bar{X}}_T \, D^{-1} \, \underline{Z} - \underline{\bar{X}}_T \, D^{-1} \, N \, D^{-1} \, \underline{Z} \\ & + \underline{\bar{X}}_T \, D^{-1} \, N \, D^{-1} \, N \, D^{-1} \, \underline{Z} - + \ldots \end{aligned} \tag{2.20}$$

If C is nonsingular the series may converge, and if, in addition, the diagonal elements of C are considerably larger than the remaining elements, then the series will converge rapidly and the high order terms may be neglected without serious error. It may be observed that for a reasonably well formulated request, the terms c_{ii} are larger than the cross-correlated terms c_{ij}. This is so since the statistic for any two given keywords to be in the same title, as measured by c_{ij}, is smaller than that for each keyword by itself, as measured by c_{ii}. Under such conditions the approximate expansion may be used.

The first order approximation of the response r is then

$$r = \sum_i \left(\frac{\bar{x}_i}{c_{ii}} - \sum_{j \neq i} \frac{\bar{x}_j}{c_{ii}} \cdot \frac{c_{ji}}{c_{jj}} \right) z_i \quad . \tag{2.21}$$

The mean square response to the collection of documents can also be expressed approximately as

$$r_m^2 = \sum_i \left(\frac{\bar{x}_i}{c_{ii}} - \sum_{j \neq i} \frac{\bar{x}_j}{c_{ii}} \frac{c_{ji}}{c_{jj}} \right) \bar{x}_i \tag{2.22}$$

and the stationary value of equation (2.11) becomes

$$\left(\frac{r_x}{r_m} \right)^2 = \sum_i \left(\frac{\bar{x}_i}{c_{ii}} - \sum_{j \neq i} \frac{\bar{x}_j}{c_{ii}} \frac{c_{ji}}{c_{jj}} \right) \bar{x}_i \quad . \tag{2.23}$$

Consequently t, the relevance measure, may be determined through equations (2.10) and (2.21).

The approximate weight for the i-th keyword may be obtained as

$$w_i = \frac{\bar{x}_i}{c_{ii}} - \sum_{j \neq i} \frac{\bar{x}_j}{c_{ii}} \cdot \frac{c_{ji}}{c_{jj}} \quad . \tag{2.24}$$

It may be seen that the contribution from the second term in the expression is small in comparison to the first term since the diagonal terms c_{ii}, c_{jj} are larger than the cross-correlated terms c_{ij}. Thus the weight for the i-th keyword may be interpreted as being approximately inversely proportional to c_{ii}, the frequency of occurrence of that word in the data base. This is to be expected since a common word does not carry too much information with it and hence has a low weight.

2.3 Flow Chart of Matched Search

The search procedure based on the above analysis involves two separate steps. For each request the entire collection is initially examined for the values of the correlations c_{ij} and the mean response r_m. It may be observed that since c_{ij} is a quantity dependent on the average properties of the whole set, a search

through the entire collection will usually be unnecessary and, in fact, impractical, for collections that contain a large quantity of documents. It may be sufficient to make this search on a representative subset rather than on the whole set. Alternatively, sampling strategy may be used on the whole set. The second search step is then initiated. This time each individual item is processed for matches with the elements in $\underline{\bar{X}}$. Values of t and r are then determined.

The search procedure may be summarized as follows:

(a) Elements of the M vectors $\underline{x}$ are merged and arranged in suitable order to form a set $\underline{\bar{X}}$.

(b) First pass: vectors of a suitably chosen subset of documents are compared with the set of $\underline{\bar{X}}$ for counts of identical keywords. The counts are required for computation of c_{ij} and r_m.

(c) Second pass: computation of r and t for each document processed.

(d) Each response is examined for its degree of relevance. Sorting may be included.

A flow chart of the procedure is shown in Fig. 2.1.

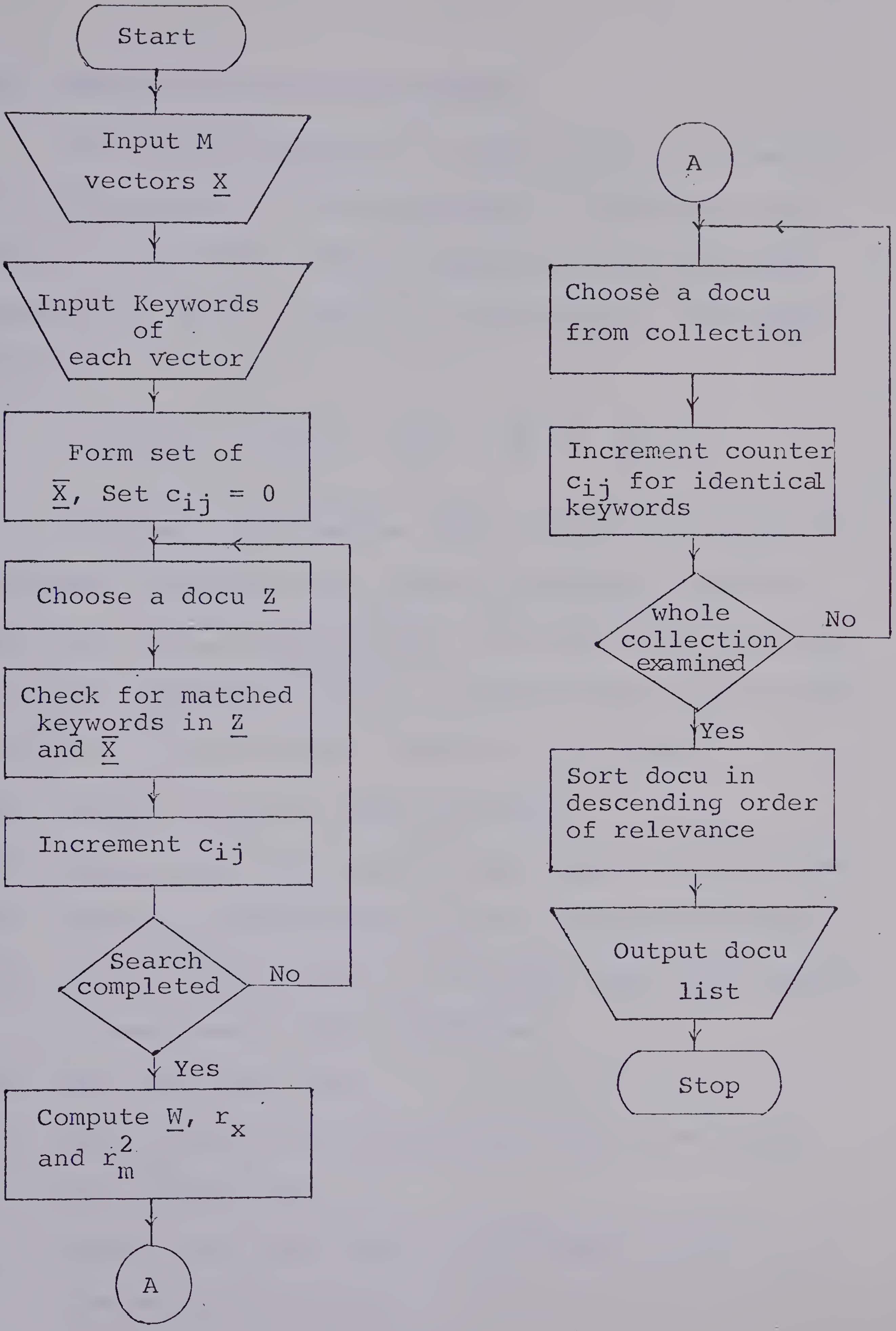
Start
Input M vectors X
Input Keywords of each vector
Form set of X̄, Set $c_{ij} = 0$
Choose a docu Z
Check for matched keywords in Z and X̄
Increment c_{ij}
Search completed
No
Yes
Compute W, r_x and r_m^2
A
A
Choose a docu from collection
Increment counter c_{ij} for identical keywords
whole collection examined
No
Yes
Sort docu in descending order of relevance
Output docu list
Stop

Fig. 2.1. Flow Chart of Matched Search

2.4 Application of matched search

The search procedure as outlined above corresponds to a one parameter form of question in which the only logic used is "OR". Thus a question containing five keywords A, B, C, D, and E is equivalent to the specification:

Profile: A OR B OR C OR D OR E .

Suppose, for example, that a request is made for documents related to the noise of engines, aircrafts and fans, and, in particular, the control and reduction of noise produced by them. A random selection from some relevant titles gives the following ten titles:

(1) Cockpit noise in small aircraft,

(2) Noise control for natural gas engine installations,

(3) Accuracy considerations in fan sound measurements,

(4) Effects of increased inlet guide vane rotor spacing on compressor noise reductions,

(5) Fans and pure tones,

(6) Noise control of on board auxiliary power units for aircraft use,

(7) Origin and characteristics of sound generation in compressors and fans,

(8) Pratt and Whitney aircraft experience in compressor noise reduction,

(9) Reduction of compressor noise radiation,

(10) Reduction of turbomachinery noise by absorption and choked flow.

A total of seven keywords were picked from these titles and the question profile was as follows:

Profile: Reduction of noise from engines, fans and aircrafts.

OR AIRCRAFT
OR COMPRESS (COMPRESSOR, COMPRESSION, ETC.)
OR CONTROL (CONTROLLER)
OR FANS
OR ENGINE
OR NOISE
OR REDUCE (REDUCTION)

The keywords are truncated to a maximum of five characters so that the use of 'COMPR' may include such words as compress, compressor and compression, among others. Arguments in favour of such word truncations have been provided by Heaps [10].

The search was made on the ACOUS tape which produced a maximum of 100 titles in decreasing order of their relevance (see Appendix B). The following list, retrieved by the approximate method,

is a subset of the output bearing the higher relevant ratios. Details of this and other examples will be discussed in a later chapter.

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	FANS, NOISE	3
.515	REDUC, AIRCR, NOISE	1
.349	NOISE, AIRCR	8
.343	AIRCR	3
.202	ENGIN, NOISE	4
.196	ENGIN	5
.172	NOISE, REDUC	10
		34 total

Titles further down in the list become less relevant to the question. This is because the majority of these contain only one matched keyword in each title. A title with only the word 'reduction' may not bear any significance to the request although the title was included in the list. It appears then that the subset, 34 in all, may be regarded as the desirable answer without much loss of important information. The cut-off value for this question may be considered to be .172 (normalized). It should be mentioned that although cut-off values may be chosen automatically, it is found that the decision

to retain whatever portion of the relevant items in the output list is quite arbitrary, and therefore should be made manually after the search is completed.

Results as obtained by the power-series approximation appear quite satisfactory. It remains to be shown that the approximate response of equation (2.21) is acceptable in terms of output response accuracy. The exact response of equation (2.13) is readily obtainable in this example for direct comparison. Results are listed below.

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	NOISE, FANS	3
.535	REDUC, AIRCR, NOISE	1
.362	NOISE, AIRCR	8
.356	AIRCR	3
.205	ENGIN, NOISE	4
.198	ENGIN	5
.178	NOISE, REDUC	10
		34 total

As can be seen by the comparison of the two outputs,agreements are excellent, exactly the same number of titles were captured and each in idential order. The approximate method results in a slightly lower value

in relevance. This is to be expected since only the first two terms of the series were retained during the analysis.

A questioner may thus use the approximation with confidence that it will produce almost as good results as obtained by the exact method. The approximation has the additional advantage of being faster and more economical in terms of computer time and storage. When the available amount of core space is limited, as would be the case in some installations, or if the question is so complex that round-off errors caused by matrix inversion become significant, then the approximation is distinctly preferable.

2.5 Effect of additional cross-correlated terms

In the above analysis the optimal weights were based on the statistical distribution of each term alone, the probable contribution from cross-correlated terms was not discussed. It may be argued that the computation of optimum weights should also be dependent, in part, on the frequency of occurrence of a given pair of keywords. It is expected, however, that whatever this contribution may be, it will only be of secondary importance.

It is thus possible to define a secondary set of weights which specifies the extent of cross-correlation between pairs of keywords. In a question containing the words A, B, C and D, the primary weights will be denoted by

$$\hat{w}_{AA},\ \hat{w}_{BB},\ \hat{w}_{CC},\ \hat{w}_{DD}$$

and the secondary weights by

$$\hat{w}_{AB},\ \hat{w}_{AC},\ \hat{w}_{AD},\ \hat{w}_{BA},\ \hat{w}_{BC},\ \text{etc.}$$

More formally let the response to the nth document be

$$\hat{r}(n) = \sum_{i} \hat{w}_{ii}\ z_{ii}(n) + \sum_{j \neq i} \hat{w}_{ij}\ z_i(n)\ z_j(n)$$

$$= \sum_{i} \sum_{j} \hat{w}_{ij}\ z_{ij}(n) \qquad (2.24)$$

where $\hat{w}_{ij}$ is the weight associated with the ith and the jth keywords in the same document and

$$z_{ij} = \begin{cases} z_i & \text{if} \quad i = j \\ \frac{1}{2} z_i z_j & \quad\ \ i \neq j \end{cases} . \qquad (2.25)$$

It is seen that z_{ij} is identical to z_{ji}, so that only half of the contribution from each term may be taken for the summation.

The mean response to the M hypothetical documents will be

$$\hat{r}_x = \frac{1}{M} \sum_i \sum_j \hat{w}_{ij} \sum_m x_{ij}(m)$$

$$= \sum_i \sum_j \hat{w}_{ij} \bar{y}_{ij} \qquad (2.26)$$

where

$$\bar{y}_{ij} = \frac{1}{M} \sum_m x_{ij}(m) \qquad (2.27)$$

and x_{ij} is specified by a relation similar to z_{ij} as expressed in equation (2.25).

The mean square response to an item in a collection of N documents will be

$$\hat{r}_m^2 = \frac{1}{N} \Sigma \hat{r}^2(n)$$

$$= \frac{1}{N} \sum_n \sum_i \sum_j \sum_k \sum_\ell \hat{w}_{ij} \hat{w}_{k\ell} z_{ij}(n) z_{k\ell}(n)$$

$$= \sum_i \sum_j \sum_k \sum_\ell c_{ijk} \hat{w}_{ij} \hat{w}_{k\ell} \qquad (2.28)$$

where

$$c_{ijk\ell} = \frac{1}{N} \sum_n z_{ij}(n) z_{k\ell}(n) \quad . \qquad (2.29)$$

Maximizing the ratio r_x^2/r_m^2 with respect to $\hat{w}_{ij}$ results in the set of equations

$$\bar{y}_{ij} - \lambda \sum_{k} \sum_{\ell} c_{ijk} \hat{w}_{k\ell} = 0 \tag{2.30}$$

for each $\hat{w}_{ij}$.

Let a sequence $\{\bar{y}_r\}$, $r=1,2,\ldots,n^2$ be defined to represent the set $\bar{y}_{11}, \bar{y}_{12}, \ldots, \bar{y}_{n1}, \ldots, \bar{y}_{nn}$; and the sequence $\{\hat{w}_s\}$, $s=1,2,\ldots, n^2$ be defined to represent $\hat{w}_{11}, \hat{w}_{12}, \ldots, \hat{w}_{n1}, \ldots, \hat{w}_{nn}$. Then for each r, equation (2.30) becomes

$$\bar{y}_r - \lambda \sum_{s=1}^{n^2} c_{rs} \hat{w}_s = 0 \tag{2.31}$$

where

$$c_{rs} = \frac{1}{N} \sum_{n} z_r(n) \, z_s(n) \quad . \tag{2.32}$$

Hence the optimum weights $\underline{\hat{W}}$ can be expressed in the following matrix equation

$$\underline{\hat{W}} = \frac{1}{\lambda} C^{-1} \, \underline{\bar{Y}} \tag{2.33}$$

and the processor response becomes

$$\begin{aligned} \hat{r}(n) &= \underline{\hat{W}}_T \, \underline{Z} \\ &= \underline{\bar{Y}} \, C^{-1} \, \underline{Z} \quad . \end{aligned} \tag{2.34}$$

The approximate response can be obtained in the form

$$\begin{aligned} \hat{r}(n) &= \underline{\bar{Y}}_T \, [D(\, I + D^{-1} N)]^{-1} \, \underline{Z} \\ &= \sum_{r} \left(\bar{y}_r \frac{1}{c_{rr}} - \sum_{s\neq r} \frac{\bar{y}_s}{c_{rr}} \cdot \frac{c_{sr}}{c_{ss}}\right) z_r \end{aligned} \tag{2.35}$$

Careful consideration of equation (2.32) reveals that the diagonal elements of C may, or may not, be larger than the other elements. Furthermore some diagonal elements may even be zero. Suppose a question consists of four keywords A, B, C and D, then the sequence $\{z_r\}_1^{16}$ may be defined as the different possible combinations of the four keywords. Thus, designating A, B, C, D as 1, 2, 3, 4 respectively

$$
\begin{aligned}
z_1 &= z_{1,1} \\
z_2 &= z_{1,2} \\
&\vdots \\
z_{12} &= z_{3,4} \\
&\vdots
\end{aligned}
$$

In general n keywords form a total of n^2 combinations and the index r is related to the indexes i and j by the expression

$$r = (i-1)\, n + j \quad . \tag{3.36}$$

Consider the case where keywords B, D never occur together in any title of the collection, then z_8 and z_{14} will be zero. Under such circumstance the response r(n) in equation (2.35) will be unbounded since $c_{8,8}$ and $c_{14,14}$ are both zero. It is therefore essential that all possible combinations of the keywords in the question exist in the data set.

The requirement of such pairing of keywords is a rather serious restriction to a user when formulating a question. It is difficult, if not impossible, to predict and use only those keywords that form pairs in the data base. Furthermore even if such a set is known, the pairs may not be all included in the question. This is because a set of n keywords will lead to a matrix of size $n^2 \times n^2$ which in most cases is not desirable in terms of the amount of computer core storage required and the length of computation time unless n is sufficiently small.

The response to the same search question as in section 2.4 is summarized below.

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	REDUC, AIRCR, NOISE	1
0.407	NOISE, FANS	3
0.378	NOISE, AIRCR	8
0.335	NOISE, REDUC	10
0.310	ENGIN, NOISE	4
0.310	NOISE, CONTR	7
0.102	AIRCR	3
		36 total

It can be seen that a slightly improved result over the previous two cases is obtained in that two more

titles were obtained. It must be mentioned however, that there are some zero diagonal elements in the matrix C. In view of this, the approximation approach was used which permitted the elimination of all those terms whose combinations do not exist in the data base. Because of this, the relevance figures quoted must be regarded as rough estimates only and not be considered as exact.

CHAPTER 3

LEAST SQUARE SEARCH

3.1 The search criterion

The previous chapter contained a description of use of the matched search to maximize a certain response with respect to the mean response to all documents. The method is based on the assumption that a 'noisy' background tends to smooth out the average properties of the data. The average total response of the collection may then be taken to be that for non-relevant titles. Consequently, a title whose response matches the question is considered to be a hit.

It is observed that the weights as indicated in equation (2.12) depend as much on the average response of all documents (as described by C) as on the form of the question vector $\underline{X}$. While the user is able to change the keywords in the question, he is unable to give his estimate of the relative importance of the keywords in relation to his question. The result may be the retrieval of some documents which seem to relate to the given vector but which are not, in fact, relevant to the question.

An alternative search procedure may be used whereby the assigned weights are dependent only on the form of the question $\underline{\bar{Y}}$ and not on the average response of the collection. In addition the user may be asked to supply his estimate of the relevance of each title in the question. To each title y(m) in $\underline{\bar{Y}}$ is attached a number f(m) to specify the relevance of the set of keywords, the title, to the question. These titles are processed with f(m), and the optimum weights will be determined from the condition that the average response is to approximate as closely as possible the estimated relevance of $\underline{\bar{Y}}$.

3.2 The search procedure

The response to a title y(m) is

$$r(m) = \sum_i w_i \, y_i(m) \tag{3.1}$$

and the error from the estimated value f(m) is

$$Q' = \sum_i w_i \, x_i(m) - f(m) \quad . \tag{3.2}$$

Suppose the question $\underline{\bar{Y}}$ consists of M titles; then the mean square error Q is given by

$$Q = \frac{1}{M} \sum_m \left[\sum_i w_i \, y_i(m) - f(m) \right]^2$$

$$= \sum_i \sum_j u_{ij} \, w_i \, w_j - 2 \sum_i v_i \, w_i + \frac{1}{M} \sum_m f^2(m) \tag{3.3}$$

where

$$u_{ij} = u_{ji} = \frac{1}{M} \sum_m y_i(m)\ y_j(m) \qquad (3.4)$$

and

$$v_i = \frac{1}{M} \sum_m f(m)\ y_i(m) \quad . \qquad (3.5)$$

u_{ij} is the cross-correlation measure between the i^{th} and the j^{th} keywords in a given title, while v_i may be considered to be the average relevance of each keyword in $\underline{\overline{Y}}$.

The response error Q is a minimum when the weights are chosen in such a way that for each w_i the following equation holds:

$$\frac{\partial Q}{\partial w_i} = 0 \qquad (3.6)$$

or

$$\sum_j u_{ij}\ w_j = v_i \quad . \qquad (3.7)$$

This results in an equation of the form

$$U\ \underline{W} = \underline{V} \quad . \qquad (3.8)$$

The set of optimum weights are obtained as

$$\underline{W} = U^{-1}\ \underline{V} \quad . \qquad (3.9)$$

When the components of $\underline{W}$ are chosen to satisfy equation (3.9) it follows from equation (3.3) that the minimum

value of Q is

$$Q_m = \frac{1}{M} \sum_m f^2(m) - \underline{V}_T\, U^{-1}\, \underline{V} \quad . \tag{3.10}$$

As in the previous chapter it proves convenient to approximate U^{-1} by a series. Thus let

$$U = G + N \tag{3.11}$$

where G is a matrix of the form

$$\begin{aligned} g_{ii} &= u_{ii} \\ g_{ij} &= 0 \end{aligned} \tag{3.12}$$

and N is a matrix whose elements are

$$\begin{aligned} n_{ij} &= u_{ij} \\ n_{ii} &= 0 \quad . \end{aligned} \tag{3.13}$$

Equation (3.10) may then be written as

$$Q_m = \frac{1}{M} \sum_m f^2(m) - \underline{V}_T (I + G^{-1} N)^{-1} G^{-1} \underline{V} \quad . \tag{3.14}$$

The expression for Q_m may be expanded into a power series with the assumption that

$$u_{ii} >> u_{ij} \tag{3.15}$$

so that high order terms may be neglected. The resulting approximate expression for Q_m is as follows:

$$Q_m = \frac{1}{M} \sum_m f^2(m) - \underline{V}_T\, G^{-1}\, \underline{V} + \underline{V}_T\, G^{-1}\, N\, G^{-1}\, \underline{V}$$

$$= \frac{1}{M} \sum_m f^2(m) - \sum_i \left(\frac{v_i}{u_{ii}} - \sum_{j \neq i} \frac{u_{ji}\, v_j}{u_{ii}\, u_{jj}}\right) v_i \tag{3.16}$$

and the weights are approximated by the expression

$$w_i = \frac{v_i}{u_{ii}} - \sum_{j \neq i} \frac{u_{ji}\, v_j}{u_{ii}\, u_{jj}} \quad . \tag{3.17}$$

Finally, the response to a general document z(n) in the collection is

$$r(n) = \sum_i w_i\, z_i(n) = \underline{V}_T\, U_T^{-1}\, \underline{Z}(n)$$

$$= V_T\, G^{-1}\, \underline{Z}(n) - V_T\, G^{-1}\, N\, G^{-1}\, \underline{Z}(n)$$

$$= \sum_i \left(\frac{v_i}{u_{ii}} - \sum_{j \neq i} \frac{u_{ji}\, v_j}{u_{ii}\, u_{jj}}\right) z_i(n) \quad . \tag{3.18}$$

3.3 The question form

As indicated in the previous section, U is a square matrix and is symmetric with respect to the diagonal elements. The search procedure then requires the question in some suitable form so that U will be well-behaved.

Suppose there are k distinct keywords in $\overline{\underline{Y}}$, and that they lead to k linearly independent columns (or rows) in U. The rank of U is then k. From elementary

matrix analysis [11], there exists at least one basic solution to equation (3.8) provided $m \geqslant k$. This leads to the following interesting result:

number of titles $\geqslant$ number of distinct keywords. (3.19)

The restriction as imposed on the least square search may not be serious to a user however. When there are many keywords in the titles of $\underline{\bar{Y}}$, a preliminary scan may be made to eliminate unimportant words. Alternatively, some keywords may be re-grouped into more hypothetical titles. In either case a question form may be obtained to satisfy (3.19).

It should be mentioned that the matrix C that appeared in the matched search was constructed from all titles of the data base and not from titles in the question. It is unlikely that any real system will have more keywords than titles. For example the Chemical Titles data base described in Chapter 5 is such that 250,000 titles contain approximately 65,000 different words and many of these words would not be used as meaningful keywords. Thus condition (3.19) will be satisfied automatically in the matched search.

A particularly simple question form results when each title in $\underline{\bar{Y}}$ contains a single keyword so that the

f(m) value associated with it may be considered as its estimated weight. In this instance the task of constructing a question is much simplified. There is the additional advantage that any desirable value may be assigned to a particular keyword and not to a collection of them as would be in the normal case. Thus the user has complete control in weighting each keyword. Subsequently U becomes a diagonal matrix with the nonzero elements equal to the estimated weights. Although the usual matrix inversion is no longer required here, it may be shown that under such circumstances the set of optimum weights will be identical to the set of f(m) values. The search procedure in essence reduces to a non-optimized one in which the weights are totally specified by the user. In either case, the question is governed by a single logic 'OR' similar to the matched search.

3.4 An example

The least square procedure differs from that of the matched search in that only one pass is needed through the data bank. Apart from this, other details follow closely the steps as outlined in section 2.3 for the matched search.

The same question for the reduction of noise from aircraft, fans, and machines may be formulated

as follows:

	Question Profile:	f(m)
OR	MACHINE, NOISE	3.00
OR	ENGINE, NOISE, CONTROL	5.00
OR	COMPRESS, NOISE, REDUCT	3.00
OR	FANS, NOISE	2.00
OR	NOISE, CONTROL, AIRCRAFT	3.00
OR	COMPRESS, FANS	2.00
OR	AIRCRAFT, COMPRESS, NOISE, REDUCT	5.00

There are seven titles in all with six distinct keywords in accordance with condition (3.19). The estimated relevances f(m) reflect the importance placed on each title. The value assigned to each title is distributed among the keywords in that title. Thus a title with a large number of keywords will normally be given a high value of relevance unless it is intended to have negative or zero weights attached to some keywords.

Results of using the least square search with the above question are tabulated below:

Table 3.1 Result of least square search

Normalized relevance	Keywords in title	Number of titles retrieved
1.00	ENGIN , NOISE	4
0.80	ENGINE	7
0.80	REDUC , AIRCR, NOISE	1
0.60	AIRCR, NOISE	8
0.40	AIRCR	3
0.40	FANS, NOISE	4
0.40	REDUC, NOISE	7
		34 total

These 34 titles are identical with those obtained by the matched search.

CHAPTER 4

TWO PARAMETER QUESTION

4.1 The search criterion

As was discussed in the matched search a question may be formed simply by direct coding of titles from actual or hypothetical documents. In the case of actual documents that are known to be relevant, only minor editing is required to remove such non-relevant words as the general 'articles' and 'prepositions'. Thus little human effort is involved. As for the least square search some care is needed to pick keywords from titles so that a well-behaved matrix is eventually obtained. A question form which needs even more human involvement will be discussed here, together with a new search procedure to deal with such a question.

The question will be supposed to consist of 2 'parameters' each of which contains a set of keywords represented by a vector. The need for a 2-parameter search may arise in situations where titles of highly specialized topics are to be retrieved. As will be seen below, a document under examination must contain keywords that match at least one keyword in each of the two parameters before the document can be considered

further for its relevance. Normally the number of hits will be reduced, and it is hoped that through the use of this procedure titles not directly relevant will be kept at a minimum in the output.

4.2 Optimization procedure

Suppose a document $z(n)$ is processed by the evaluation of two parameters given by

$$r^{(1)}(n) = \sum_i w_i^{(1)} z_i(n)$$

and

$$r^{(2)}(n) = \sum_j w_j^{(2)} z_j(n) \qquad (4.1)$$

where $r^{(1)}$ and $r^{(2)}$ are the responses of the document to the two sets of keywords in the first and second parameter respectively. The summation in i is over all keywords associated with the set belonging to the first parameter, and the summation in j is over all keywords associated with the second parameter.

Let the processor response to the n^{th} document be

$$R(n) = r^{(1)}(n) \, . \, r^{(2)}(n) \quad . \qquad (4.2)$$

$R(n)$ will be nonzero if, and only if, both $r^{(1)}(n)$ and $r^{(2)}(n)$ are nonzero.

The mean response of the processor to the collection of N titles is

$$R_m = \frac{1}{N} \sum_n R(n)$$

$$= \frac{1}{N} \sum_n r^{(1)}(n)\ r^{(2)}(n)$$

$$= \sum_i \sum_j c_{ij}\ w_i^{(1)}\ w_j^{(2)} \qquad (4.3)$$

where

$$c_{ij} = \frac{1}{N} \sum_n z_i(n)\ z_j(n) \qquad (4.4)$$

and the W's are the weights associated with the two parameters. It should be noted that c_{ij} will not, in general, equal c_{ji} since the indices i and j range over two different sets of keywords. The sets may, or may not, have some keywords in common.

Suppose the question is denoted by the two vectors $x^{(1)}$ and $x^{(2)}$; the system response is then

$$R_x = r^{(1)} \cdot r^{(2)}$$

$$= \sum_i \sum_j w_i^{(1)}\ w_j^{(w)}\ x_i^{(1)}\ x_j^{(2)}\ . \qquad (4.5)$$

It will be assumed that the retrieved criterion requires that R_x is to be a maximum in comparison to R_m. The condition for the response R_x to be stationary

with respect to the mean response R_m is that $w_i^{(1)}$ and $w_j^{(2)}$ be chosen to satisfy the equations

$$\frac{\partial}{\partial w_i^{(1)}} \left(\frac{R_x}{R_m}\right) = 0$$

and

$$\frac{\partial}{\partial w_j^{(2)}} \left(\frac{R_x}{R_m}\right) = 0 \quad . \tag{4.6}$$

These result in two sets of equations

$$x_i^{(1)} \sum_j w_j^{(2)} x_j^{(2)} - \lambda \sum_j c_{ij} w_j^{(2)} = 0, \quad \text{for each } i \tag{4.7}$$

and

$$x_j^{(2)} \sum_i w_i^{(1)} x_i^{(1)} - \lambda \sum_i c_{ij} w_i^{(1)} = 0, \quad \text{for each } j \tag{4.8}$$

where λ denotes the stationary value of R_x/R_m.

Equation (4.7) may be rearranged into

$$x_i^{(1)} = \frac{\lambda \sum_j c_{ij} w_j^{(2)}}{\sum_j w_j^{(2)} x_j^{(2)}} \tag{4.9}$$

The denominator is independent of the index j and thus it proves convenient to introduce new variables $v_j^{(2)}$ defined by

$$v_j^{(2)} = \frac{\lambda}{\sum_j w_j^{(2)} x_j^{(2)}} \cdot w_j^{(2)} = \lambda^{(2)} w_j^{(2)} \quad . \tag{4.10}$$

Similarly, from equation (4.8) the variables $v_i^{(1)}$ may be defined by

$$v_i^{(1)} = \frac{\lambda}{\sum_i w_i^{(1)} \; x_i^{(1)}} \cdot w_i^{(1)} = \lambda^{(1)} \; w_i^{(1)} \; . \qquad (4.11)$$

The set of equations then becomes

$$\sum_j c_{ij} \; v_j^{(2)} = x_i^{(1)} \qquad \text{for each } i \qquad (4.12)$$

and $$\sum_i c_{ij} \; v_i^{(1)} = x_j^{(2)} \qquad \text{for each } j \; . \qquad (4.13)$$

Thus in matrix form,

$$C \; \underline{V}^{(2)} = \underline{X}^{(1)} \qquad (4.14)$$

$$C_T \underline{V}^{(1)} = \underline{X}^{(2)} \; . \qquad (4.15)$$

As indicated in equations (4.10) and (4.11), the V's are directly related to the W's, and apart from a constant of proportionality the two equations have identical form. Subsequently the V's may be considered as replacing the W's as the set of optimum weights determined by the optimization process.

The optimum weights are found from the equations

$$V^{(1)} = C_T^{-1} \; \underline{X}^{(2)} \qquad (4.16)$$

and $$V^{(2)} = C^{-1} \; \underline{X}^{(2)} \qquad (4.17)$$

4.3 Question formulation

As has been pointed out previously C will not be a symmetric matrix, and for the existence of C^{-1}, it is necessary that the two parameters contain the same number of keywords. This does not seem to be a serious restriction since the parameters may be extended to meet this requirement by the addition of extra keywords. Alternatively some keywords in one parameter may be included to the other set. Thus some care should be taken in selecting keywords for use in a question.

Once a group of keywords has been chosen to represent a question, it is necessary to separate them into two sets to fit the search criterion. Normally it will be desirable for keywords of a similar nature to be placed in the same set. An example is provided by words that are synonyms. When these are combined with keywords in the other set there result word pairs of similar meaning. The use of synonyms in searching has been found to be beneficial, since many relevant items will not otherwise be retrieved.

Another fact that may be taken into consideration is the fact that many words tend to associate highly with certain other words. If such associated word pairs are included in the question it may be advisable not to

put them in the same set since words in the same parameter are not included in the computation of C.

The matrix C will not in general be unique for a given question. It is clear that C^{-1} may have different values depending on the manner in which the keywords are arranged in the two parameters. In most cases interesting variations in C may be obtained by the interchange of keywords between the two sets. Other possible variations result from the interchange of keywords within the same set. It should be mentioned that given a group of keywords, it is generally not known, a priori, how to construct the "best" question from these keywords.

In contrast to either the matched search or the least square search, the question logic involves 'AND' and 'OR'. Thus a question for the 2-parameter search has the following form:

Parameter 1 : A OR B OR C

AND Parameter 2 : D OR E OR F

4.4 An example

The two parameter search requires two passes through the data base, the first time for the collection of statistical count of the keywords in the question,

and the second time for matching keywords in titles with those in the question. Except for some minor changes in detail, the procedural steps as outlined in the flow-chart for the matched search may be applied here.

A possible query consisting of six keywords may be constructed from the original set of titles for the search of noise from machine, aircraft and fans. The keywords are arranged in the two parameters as follows :

Parameter 1 : AIRCR ENGIN REDUC

Parameter 2 : ACOUS NOISE SOUND

Results are shown in Table 4.1.

Table 4.1 Results of 2-parameter search

Normalized relevance	Keywords in title	Number of documents retrieved
1.000	SOUND, ENGIN , NOISE	1
.650	SOUND, ENGIN	3
.350	ENGIN , NOISE	3
.180	SOUND, AIRCR	1
.105	AIRCR, NOISE	7
.100	ACOUS, ENGIN	3
.075	ACOUS, AIRCR, NOISE	1
.070	NOISE, REDUC	8
.050	REDUC, ACOUS, NOISE	1
.050	REDUC, AIRCR, NOISE	1
		29 total

The five titles that were missed were the following:

(1) Noise of centrifugal fans.

(2) Measurement of noise from ducted fans.

(3) Real noise originated from power speed ducted centrifugal fans.

(4) Hearing of naval aircraft maintenance personnel.

(5) Further experiments judging noisy aircraft flight.

CHAPTER 5

COMPARISON AND CONCLUSION

5.1 Comparative study of search procedures

The previous chapters were devoted to the analysis of different optimum criteria for document search, with results of an example given for each method. This example was more for the purpose of illustration than for comparative studies. With more search questions it will be possible to obtain evidence of advantages and drawbacks of each criterion by analyzing a variety of results. It may also be possible to reach a conclusion to indicate the preferable choice among the four procedures.

In the following discussion the exact solution of the matched criterion, instead of the approximate version, will be used for analysis and comparison with other criteria. This is because the examples in the present investigation involve a small number of keywords and the former version is then capable of handling such questions efficiently in terms of computer time and storage. As has been mentioned previously, whenever a question requires lengthy computing time the approximate method proves to be more economical and convenient to use.

All desirable output responses are trimmed and tabulated according to the following scheme. The titles are sorted in descending order of their total relevances. These values are normalized as fractions in relation to the largest value which appears at the top of the list and which becomes equal to 1. To indicate the nature of each title retrieved, an entry is provided to list the keywords in it and contained in the question. In the following questions, titles were taken from the Cumulative Index of the *Journal of the Acoustical Society of America*.

Question A

In an attempt to search for articles that relate to the design of auditoria, or music halls and rooms, or studios, and with emphasis placed on sound absorption and reverberation, ten relevant titles were selected for use as a question for the matched search. These titles are listed below. They contain a total of 26 distinct keywords as underlined, and resulting hits are summarized in Table 5.1.

The titles are the following:

(A1) *Sound* *amplification* in *reverberant* *spaces*.

(A2) *Musical*-*acoustical* *criteria* of *concert* *halls*.

(A3) *Models* as an aid in the *acoustic* *design* of *auditoria*.

(A4) The *acoustic* *design* of *studio* and listening *rooms*.

(A5) *Acoustics* for *music* *performance*.

(A6) *Formation* and *testing* of *acoustic* *conditions* in *rooms*.

(A7) *Acoustic* *considerations* in the *design* of *recording* *studios*.

(A8) The *design* of small *studios* and listening *rooms*.

(A9) The *effect* of *absorbent* *walls* of a *closed* *space* on the *uniformity* of the *sound* *field*.

(A10) The *acoustic* *conditions* accepted by listeners in an *auditorium*.

Table 5.1 Matched Search A

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	ACOUS, CRITE, CONCE, HALLS, PERFO, MUSIC	1
0.994	DESIG, CONCE, HALLS	1
0.921	ACOUS, STUDI, ROOMS, MUSIC	1
0.710	MUSIC, ROOMS	1
0.700	SOUND, CLOSE, ROOMS	1
0.673	ACOUS, HALLS	1
0.670	REVER, ROOMS, HALLS	1
0.660	ABSOR, HALLS	1
0.639	REVER, HALLS	1
0.616	CRITE, ACOUS, ROOMS	1
0.536	ACOUS, ROOMS	1
0.533	ACOUS, FIELD, ROOMS	2
0.526	EFFEC, DESIG, CONSI	1
0.521	SOUND, ABSOR, ROOMS	1
0.519	ROOMS	6
0.502	REVER, ROOMS	4
0.425	STUDI, ACOUS, CONSI	1
0.405	MODEL, ACOUS, DESIG, AUDIT	1
0.384	ACOUS, DESIG, PERFO	1
0.381	ACOUS, DESIG, AUDIT	3
0.360	AUDIT, DESIG	1
		32 total

Cut-off at 0.360

For the cross-correlated matched search, out of the 10 original titles 7 were selected (A2, A3, A4, A5, A7, A8 and A9) containing 9 distinct keywords to form the question. They were: ABSOR, AUDIT, CONCE, DESIG, HALLS, MUSIC, REVER, ROOMS, STUDI. Results were:

Table 5.2 Cross-correlated matched search A

Normalized relevance (N.R.)	Keywords in title	Number of documents retrieved
1.000	CONCE, HALLS, MUSIC	1
0.976	DESIG, CONCE, HALLS	1
0.908	STUDI, ROOMS, MUSIC	1
0.718	ABSOR, AUDIT	1
0.595	CONCE, AUDIT	1
0.554	ABSOR, CONCE	1
0.495	ABSOR, REVER	5
0.477	DESIG, AUDIT	5
0.445	ABSOR, HALLS	1
0.431	ABSOR, ROOMS	1
0.313	DESIG, CONCE	1
0.274	STUDI, REVER	1
0.274	MUSIC, ROOMS	1
0.269	REVER, HALLS	1
0.255	REVER, ROOMS	6
0.034	ROOMS	10
		38 total

Cut-off at 0.034.

In the least square method 9 titles (all except A9) and 8 keywords were used, and the question profile was chosen as follows:

	f(m)
REVER, AUDIT	4.0
MUSIC, HALLS	6.0
DESIG, AUDIT	4.0
DESIG, STUDI, ROOMS	6.0
MUSIC	4.0
DESIG, STUDI	4.0
STUDI, ROOMS	4.0
CONCE, HALLS	3.0
AUDIT	1.0

Table 5.3 Least Square Search A

Normalized relevance	Keywords in title	Number of titles retreieved
1.000	STUDI, ROOMS, MUSIC	1
0.902	CONCE, HALLS, MUSIC	1
0.806	MUSIC, ROOMS	1
0.710	DESIG, CONCE, HALLS	1
0.678	DESIG, REVER	1
0.645	REVER, ROOMS	4
0.613	REVER, HALLS	1
0.550	REVER, STUDI	1
0.484	DESIG, AUDIT	5
0.452	DESIG, CONCE	2
0.355	REVER	2
		20 total

Cut-off at 0.355

The question for the 2-parameter search was arranged into the form

P1 : AUDIT, ROOMS, HALLS, STUDI
P2 : CONCE, MUSIC, PERFO, REVER.

A total of five titles were produced as hits; they are listed in table 5.4.

Table 5.4 2-parameter search A

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	PERFO, STUDI	1
0.666	CONCE, HALLS, PERFO, MUSIC	1
0.333	STUDI, REVER	1
0.333	REVER, HALLS	1
0.333	STUDI, MUSIC, ROOMS	1
		5 total

Cut-off at 0.333

Question B

This request is concerned with the horn type loudspeakers of a stereo playback system. The following ten titles form the basis of the question.

(B1) Evaluation of a *modulated* air-flow *loudspeaker*.

(B2) Evaluation of a *stereophonic* *loudspeaker* by *multiple* *microphone* arrays.

(B3) Method of improving *acoustic* *transmission* in folded *horns*.

(B4) *Generation* of intense *audio* *sound* *fields* utilizing arrays of *multiple* *driver* *horns*.

(B5) New *stereophonic* projection *console*.

(B6) Theory of isophonic *speaker* *systems*.

(B7) Experimental full range *electrostatic* *speaker*.

(B8) *Crossover* problems in *two* way *loudspeaker* *systems*.

(B9) *Infinite* *baffle* comprising a spherical shell of foamed plastic.

(B10) Improved *directional* *distribution* from *horn* arrays.

The underlined keywords were used for the matched search, while the cross correlated version retained only nine keywords in the same number of titles, namely, AUDIO, BAFFL, HORNS, HORN, LOUDS, MODUL, SPEAK, STERE, SYSTE. Results are as shown in Tables 5.5 and 5.6 respectively.

Table 5.5 Matched Search B

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	HORN, SYSTE, HORNS	1
0.920	STERE, LOUDS	1
0.728	AUDIO, SOUND, FIELD, MULTI, DRIVE, HORNS	1
0.564	SPEAK, SYSTE	1
0.558	ELECT, SPEAK	1
0.558	SPEAK	3
0.506	HORN	4
0.484	HORNS	6
0.484	ACOUS, TRANS, HORNS	1
0.476	HORNS, TRANS	1
0.472	ACOUS, HORNS, SOUND	1
0.470	LOUDS, MODUL	3
0.461	LOUDS, SYSTE	1
0.460	NOISE, TWO, LOUDS, SYSTE	1
0.460	ELECT, LOUDS	1
0.449	LOUDS, NOISE	1
0.445	TWO, LOUDS	3
		31 total

Cut-off at 0.445

Table 5.6 Cross-correlated search B

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	HORN, SYSTE, HORNS	1
0.366	LOUDS, SYSTE	2
0.342	STERE, SYSTE	1
0.326	SPEAK, SYSTE	1
0.134	LOUDS, MODUL	3
0.120	STERE, LOUDS	1
0.116	MODUL, AUDIO	3
0.090	AUDIO, HORNS	1
0.031	HORN	4
0.021	STERE	5
0.020	LOUDS	13
0.016	HORNS	9
0.013	SPEAK	4
		48 total

Cut-off at 0.013

Titles B1, B2, B3, B4, B5, B6, B9 and B10 containing a total of eight distinct keywords were chosen for the least square search. These keywords together with the corresponding relevance estimates are as follows:

LOUDS	3.0
STERE, LOUDS	5.0
HORNS	2.0
DRIVE, HORNS	3.0
STERE, CONSO	3.0
SPEAK	2.0
BAFFL	2.0
HORN	2.0

This results in 42 titles as in the following table:

Table 5.7 Least square search B

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	STERE, LOUDS	1
0.800	HORN, HORNS	1
0.600	DRIVE, HORNS	1
0.600	LOUDS	18
0.600	DRIVE, BAFFL	2
0.400	HORNS	9
0.40	HORN	3
0.40	SPEAK	4
0.40	BAFFL	3
		42 total

Cut-off at 0.40

Another set of 8 keywords was selected for the 2-P search. They were specified as

P1 : LOUDS, MULTI, INFIN, DRIVE

P2 : SYSTE, MODUL, HORNS, STERE

Table 5.8 summarizes the results.

Table 5.8 2-P Search B

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	STERE, LOUDS, MULTI	1
1.000	MULTI, DRIVE, HORNS	1
0.600	MULTI, SYSTE	3
0.533	INFIN, SYSTE	2
0.60	LOUDS, SYSTE	2
0.20	LOUDS, MODUL	3
0.15	DRIVE, MODUL	1
		13 total

Cut-off at 0.15

Question C

It will be supposed that the required documents relate to the effect of ultrasonic waves in light beams, with attention focused on the amplitude, reflection or diffraction variation of the latter. Ten titles

were selected as the basic reference with a total of twenty basic keywords used in the matched search. They are as follows:

(C1) Diffraction of light by two ultrasonic waves of the same frequency.

(C2) Diffraction of electromagnetic waves by sound waves.

(C3) Optical effects of ultrasonic waves producing phase and amplitude modulation.

(C4) Determination of finite amplitude distortion by light diffraction.

(C5) Effect of ultrasonic waves on concentration gradients.

(C6) Optical methods for the measurement of sound pressure in liquids.

(C7) Determination of finite amplitude distortion by light diffraction.

(C8) Electromagnetic reflection from sound waves.

(C9) Diffraction of light by ultrasonic waves of various standing wave ratios.

(C10) Diffraction of wide and narrow light beams by distorted finite amplitude progressive ultrasonic waves in water.

Results of this search appear in Table 5.9.

Table 5.9 Matched Search C

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	DIFFR, LIGHT, ULTRA, RATIO	1
0.905	DIFFR, LIGHT, BEAMS, DISTO, FINIT, AMPLI, ULTRA, WATER, DIFFR	1
0.798	LIGHT, ULTRA, BEAMS, FREQU	1
0.784	FINIT, AMPLI, DISTO, LIGHT, DIFFR	1
0.780	ULTRA, LIGHT, DIFFR	1
0.756	DIFFR, LIGHT, AMPLI, ULTRA	2
0.750	DIFFR, LIGHT, ULTRA, DISTO, LIQUI	1
0.744	LIGHT, DIFFR, DISTO, ULTRA	1
0.723	ULTRA, DIFFR, LIGHT	2
0.720	ULTRA, DIFFR, LIGHT, FINIT, AMPLI	2
0.695	EFFEC, LIGHT, DIFFR, ULTRA	2
0.692	DIFFR, LIGHT	2
0.688	DIFFR, LIGHT, ULTRA	8
		25 total

Cut-off at 0.688

With the cross-correlated search, the following 9 keywords were used: AMPLI, BEAMS, DIFFR, ELECT, LIGHT, MAGNE, OPTIC, REFLEC, ULTRA. Results are presented as follows:

Table 5.10 Cross-correlated search C

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	DIFFR, LIGHT, BEAMS, AMPLI, ULTRA	1
.675	LIGHT, DIFFR, AMPLI, ULTRA	4
.633	DIFFR, LIGHT, ULTRA, BEAMS	1
.362	AMPLI, LIGHT, DIFFR	1
.355	LIGHT, DIFFR, ULTRA	17
.333	DIFFR, ULTRA	1
.320	ULTRA, BEAMS, DIFFR	1
.302	AMPLI, ULTRA, OPTIC	4
.290	OPTIC, MAGNE, ULTRA	1
.286	OPTIC, ULTRA	1
.286	ULTRA, BEAMS	1
		33 total

Cut-off at .286

The question for the least square search was chosen as follows:-

DIFFR, LIGHT	3.0
DIFFR, SLECT	3.0

OPTIC, ULTRA, AMPLI	5.0
AMPLI, LIGHT, DIFFR	5.0
ULTRA	1.0
OPTIC	2.0
ELECT, MAGNE	3.0
DIFFR, ULTRA	3.0
DIFFR, LIGHT, BEAMS, ULTRA	6.0

It is made up of 9 titles and 8 keywords.

Table 5.11 Least square search C

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	DIFFR, LIGHT, BEAMS, AMPLI, ULTRA	1
.75	ULTRA, LIGHT, DIFFR, BEAMS	1
.75	DIFFR, LIGHT, AMPLI, ULTRA	4
.625	OPTIC, MAGNE, ULTRA	1
.625	ULTRA, BEAMS, DIFFR	1
.625	OPTIC, AMPLI, ULTRA	1
.625	AMPLI, LIGHT, DIFFR	1
.625	LIGHT, DIFFR, ULTRA	17
		31 total

Cut-off at .625

The 2-P search used the following question:

P1 : LIGHT, OPTIC, ULTRA, ELECT

P2 : BEAMS, AMPLI, DIFFR, PRESS

Table 5.12 2-P search C

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	ULTRA, BEAMS	1
0.864	ULTRA, BEAMS, DIFFR	1
0.730	OPTIC, DIFFR	3
0.574	LIGHT, BEAMS	1
0.476	OPTIC, PRESS	1
0.394	DIFFR, LIGHT, BEAMS, AMPLI, ULTRA	1
0.368	DIFFR, ULTRA, BEAMS	1
		9 total

Question D

The query is made for recording techniques with magnetic tapes or disks. With a set of the following 8 titles, the matched search produced 20 relevant documents as in Table 5.13.

(D1) Stereophonic and quasi-stereophonic reproduction.

(D2) Improved quasi-stereophonic and colorless artificial reverberation.

(D3) Analysis of a piezoelectric phonograph transducer as a function of the clamping position.

(D4) Surface and groove noise in disk recording media.

(D5) Determination of the recording performance of a tape from its magnetic properties.

(D6) Experimental and theoretical investigation of the magnetic properties of iron oxide recording tape.

(D7) Method and apparatus for dual sound track recording.

(D8) Method and apparatus for magnetic recording and reproducing.

Table 5.13 Matched search D

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	RECOR, TAPE, IRON, OXIDE	1
0.699	SURFA, GROOV, NOISE, DISK, RECOR, MUSIC	2
0.608	MAGNE, RECOR, TAPE	1
0.594	RECOR, PERFO, TAPE	1
0.580	TAPE, RECOR	1
0.574	SOUND, TAPE, RECOR	1
0.482	PIEZO, PHONO, TRANS	1
0.465	PHONO	3
0.448	STERE, SOUND, REPRO	3
0.410	MAGNE, TAPE	1
0.377	TRACK	4
		20 total

Cut-off at 0.377

With the cross-correlated search, the following 8 keywords from the 8 titles were used:

DUAL, GROOV, PHONO, RECOR, REPRO, STERE,
TAPE, TRACK.

Table 5.14 Cross-correlated search D

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	RECOR, TAPE	5
0.902	GROOV, RECOR	2
0.279	TAPE	1
0.257	STERE, REPRO	3
0.222	GROOV	1
0.0159	REPRO	5
0.0102	STERE	11
0.0047	RECOR	5
		33 total

The search question for the least square method was as follows:

STERE, REPRO	2.0
STERE, DISK	2.0
PHONO, DISK	2.0
GROOV, DISK, RECOR	3.0
RECOR, TAPE	2.0
RECOR	1.0

RECOR, REPRO	2.0
STERE, TAPE	2.0

These contain 7 keywords in the 8 titles, the results of which are in Table 5.15.

Table 5.15 Least square search D

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	GROOV, DISK, RECOR	2
0.667	RECOR, TAPE	1
0.667	STERE, REPRO	3
0.667	RECOR, TAPE	4
0.667	DISC, RECOR	1
0.333	PHONO	4
0.333	GROOV	1
0.333	STERE	1
0.333	RECOR	3
		20 total

As for the 2-P search, the 2 vectors were as follows :

P1 : DISK, GROOV, RECOR, NOISE

P2 : SURFA, TAPE, MAGNE, RECOR

Table 5.16 2-P search D

Normalized relevance	Keywords in title	Number of titles retrieved
1.000	SURFA, GROOV, NOISE, DISK, RECOR	2
0.464	NOISE, RECOR, TAPE	1
0.400	MAGNE, RECOR	1
0.392	MAGNE, NOISE	1
0.328	RECOR, TAPE	4
0.072	RECOR	6
		15 total

5.2 Discussion and conclusion

Although only 4 examples were analyzed in the previous section, a general pattern may be recognized from the results in that the cross-correlated version produces the most relevant items, followed by the least square and the matched methods, while the 2-parameter produces the least number.

In general the matched search provides an overall unsophisticated approach to document searching since, with the minimum effort on the part of the questioner, useful responses may be obtained. In this respect, it may be regarded as a tool for general purpose searches which do not require very critical results.

Although many drawbacks, such as the mathematical complexity, large computer storage and longer computing time involved, have been observed in the more refined matched search with cross-correlated terms, responses obtained were very favourable and, in spite of some very crude estimates of numerical values, results were generally better than without the use of the extra terms in regard to Recall and Precision. It must be mentioned, however, that the choice of keywords to make up the question was partly based on the questioner's familiarity with the literature in the data base. Consequently the success of the search was, to some extent, affected by it. The same success may or may not be obtained by another user with little or no experience in question formulations or with little familiarity with the base keywords in the collection.

As can be seen from the results, the 2-parameter criterion may be considered a rather specialized method. Results were, in some cases, too restricted for it to be regarded useful as a general purpose search procedure. The following example will illustrate this fact. One of the methods currently being used by the Alberta Information Retrieval Association is a modified version of the 2-parameter procedure where the question may be specified by many, instead of only two, parameters. The

logics used may be 'AND', 'OR', 'NOT', or 'NOR'. It also differs from the 2-parameter approach in that the weights are not optimized as was done in Chapter four.

A user submits his question for searching on the fortnightly Chemical Titles tapes over a period of time. Results on each search are returned to the user, and he is asked to indicate the titles that he regards as relevant. A data base based on a number of Chemical Titles issues is constructed, and weights are determined by word counts made on those words that appeared in the relevant titles. Details on Chemical Titles search may be found elsewhere [12] , the following question may serve as an example:

Profile : Peptide Synthesis
P1 : PETIDE, PROTEIN
P2 : SYNTHE, DERIV, PREPARA .

From the accumulated lists of 27 relevant titles, the following words were found to be most satisfactory in obtaining results relevant to the original question.

Normalized weight	Word
1.000	DES
.523	PEPTIDE
.500	PROLYL
.455	LYSYL

Normalized weight	Word
.385	ORNITHINE
.261	GLYCYL
.156	THREON
.135	SERINE
.134	TERMIN
.132	LYSINE
.064	SYNTHE
.045	PROTEIN

While PEPTDE is considered an important word, the original words DERIV and PREPARA are not in this list. This indicates that much better results may be obtained when the above list of words are used as a question for the matched search.

As shown in the four examples the least square procedure can provide very satisfactory results. Success also depends, to some extent, on the experience of the user for, by manipulating the relevance estimates f(m), emphasis may be placed on some preferred keywords. In most cases, this results in marginally improved responses. It may then be speculated that an experienced user would prefer to use this method.

As has been observed in section 3.3, a degenerate form of the least square method exists. In this case, a user may specify an estimated value of relevance for

each keyword separately. This results in a diagonal matrix, and the computed weights will be equal to the originally specified values of relevance. Since the weights are not optimized the response will not be optimum. In spite of this, a user may sometimes find it useful to use this form of search in order to place emphasis on certain keywords.

A factor that must concern the user is cost. It is hoped that the optimum procedures discussed here will lead to solution of some of the problems in getting the best response efficiently and economically. Questions involving computing time and machine storage, as well as the question form, are responsible for the total cost. In regard to this, some investigations have been done in relation to Chemical Titles searching using the matched criterion [12]. At present no such cost functions are available for either the 2-parameter or the least square procedures. Until the time that these are established, the least square criterion seems to offer a more satisfactory procedure for document search.

REFERENCES

(1) Borke, H., Burnaugh, H.P. "Interactive Displays for Document Retrieval", Information Display, Vol. 3, pp. 47-90, 1966.

(2) Drew, D.L., Summit, R.K., Tanaka, R.I., Whiteley, R.B. "An On-Line Technical Library Reference Retrieval System", American Documentation, Vol. 17, pp. 3-7, 1966.

(3) Salton, G. "A Document Retrieval System for Man-Machine Interaction", Association for Computing Machinery, Proceedings of the 19th Annual Conference, Philadelphia, 1964.

(4) Salton, G. "The SMART System - Retrieval Results and Future Plans", Scientific Report No. ISR - 11 to the National Science Foundation, Vol. 1, Section 1, Cornell University, Dept. of Computing Sc., 1966.

(5) Salton, G. Automatic Information Organization and Retrieval, McGraw Hill, New York, 1968.

(6) Barkle, J.K. "Construction of Weighted Term Profiles by Measuring Frequency and Specificity in Relevant Items", 2nd International Conference on Mechanized Information Storage and Retrieval, Cranfield, Bedfordshire, England, September 1969.

(7) Wiener, N. Extrapolation, Interpolation, and Smoothing of Stationary Time Series, J. Wiley & Sons, Inc., New York, 1949.

(8) Levinson, N. "The Wiener RMS Error Criterion in Filter Design and Prediction", Journal of Mathematics and Physics, Vol. 15, No. 4, pp. 261-278, January 1947.

(9) Dwork, B.M. "Detection of A Pulse Superimposed on Fluctuation Noise", Proceedings of IRE, pp. 771-774, July 1950.

(10) Heaps, H.S. "Boolean, Fractional, and Association Searches on Truncated Title Words", Proceedings of the American Society for Information Science, Columbus, Ohio, October 1968.

(11) Perlis, S. Theory of Matrices, Addison-Welsely Press Inc., Cambridge, Mass., 1952.

(12) Heaps, H.S. and Thiel, L.H. "Optimum Procedures for Economic Information Retrieval", Information Storage and Retrieval, Vol. 6, 1970. In press.

APPENDIX A

Program Listing

The following pages contain source programs of the matched, the cross-correlated, the least square, and the 2-parameter search.

In the matched and the least square search the subroutine FINVER is used to compute the inverse of a matrix using an existing library subprogram CS011A. A different subroutine FINVA is used in the 2-parameter search.

```
C
C
C
C     *** MATCHED SEARCH ***
C
      COMMON WTT(200),IWT(200),C
      REAL MAT(60,60),INVERS(60,60),RHS(60),BB(60,60),RR(60),XX(60)
      INTEGER PIVOT(60),C(100,100)
      DIMENSION NOTERM(20),KEYWD(20,20),NUMI(20),KOUNT(200),ITITL(100),
     1ITEM(100,50),STORE(100),KEYBAR(200)
      DATA KBLAN/'    '/,KZZZZ/'ZZZZ'/,KLBR/'(   '/,KRBR/')   '/,
     1KCOM/',   '/
C
C     READ SPECIFIED KEYWORDS KEYWD(I,M) M=1 TO KAPM
C     FIND NUMI(M)=2*NUMBER OF KEYWORDS KEYWD(I,M)
C
   10 READ(5,21,END=556) KAPM,ITRACE
   21 FORMAT(2I4)
      NUM=0
      DO 35 M=1,KAPM
      I1=1
   24 I2=I1+25
      READ(5,25)(KEYWD(I,M),I=I1,I2)
   25 FORMAT(13(A4,A2))
      I3=I2-1
      DO 28 I=I1,I3,2
      IF(KEYWD(I,M).EQ.KBLAN) GOTO 30
   28 CONTINUE
      I1=I1+26
      GOTO 24
   30 K=I-1
      WRITE (6,333)
  333 FORMAT(/5X,'THE FOLLOWING ARE KEYWORDS FROM A QUESTION')
      WRITE(6,334) (KEYWD(I,M),I=I1,K)
  334 FORMAT (15X,2A4)
      NOTERM(M)=K/2
      WRITE(6,555) NOTERM(M)
  555 FORMAT (5X,'NO. OF TERMS=',I2)
      NUMI(M)=K
      NUM=NUM+NUMI(M)
   35 CONTINUE
      IF(NUM.LE.200) GOTO 37
      WRITE(6,36)
   36 FORMAT(5X,' MORE THAN 100 KEYWORDS READ ')
      STOP
   37 IF(ITRACE.NE.1) GOTO 40
      WRITE(6,38) KAPM
   38 FORMAT(5X,' KAPM=',I4)
      WRITE(6,39) NUM
   39 FORMAT(5X,'NUM=',I4/)
C
C     COMBINE KEYWORDS TO FORM KEYBAR(I) IN ALPHABETIC ORDER
C     WITH WEIGHTS IWT(I) I=1 TO NUM. FIRST MERGE KEYWDS
C
   40 I1=1
```

```
   44 DO 50 M=1,KAPM
      I2=I1+NUMI(M)-1
      DO 48 I=I1,I2
      KEYBAR(I)=KEYWD(I-I1+1,M)
   48 CONTINUE
      I1=I2+1
   50 CONTINUE
      NUM=I1-1
C     SET IWT(I)=1 FOR I=1,NUM-1 IN STEPS OF 2
      J=NUM-1
      DO 55 I=1,J,2
      IWT(I)=1
   55 CONTINUE
C
C     BUBBLE SORT KEYBAR(I)
      DO 78 K=1,J,2
      I=NUM-1
      LAST=0
   62 IF(KEYBAR(I) - KEYBAR(I-2)) 66,64,76
   64 IF(KEYBAR(I+1)-KEYBAR(I-1)) 66,70,76
   66 IT=KEYBAR(I)
      KEYBAR(I)=KEYBAR(I-2)
      KEYBAR(I-2)=IT
      IT=KEYBAR(I+1)
      KEYBAR(I+1)=KEYBAR(I-1)
      KEYBAR(I-1)=IT
      IT=IWT(I)
      IWT(I)=IWT(I-2)
      IWT(I-2)=IT
      LAST=1
      GOTO 76
   70 IWT(I-2)=IWT(I-2)+IWT(I)
      IWT(I)=0
      KEYBAR(I)=KZZZZ
      KEYBAR(I+1)=KBLAN
      LAST=1
   76 I=I-2
      IF(I.GT.K) GOTO 62
      IF(LAST.EQ.0) GOTO 80
   78 CONTINUE
C
C     ELIMINATE DUPLICATE KEYWORDS
C
      DO 79 I=1,J,2
      IF (KEYBAR(I).EQ.KZZZZ) GOTO 77
   79 CONTINUE
      GOTO 80
   77 NUM=I-1
   80 IF(ITRACE.NE.1) GOTO 100
      WRITE(6,39) NUM
      NA=NUM-1
      WRITE(6,90)((I,KEYBAR(I),KEYBAR(I+1),KLBR,IWT(I),KRBR,KCOM),
     AI=1,NA,2)
   90 FORMAT (5(5X,I3,1X,2A4,1X,A1,I2,2A1)/)
      NUMI(M)=I-1
```

```
C
C     SET C(I,J)=0 FOR I,J=1 TO NUM
C
      NN=NUM/2
  100 DO 102 I=1,NN
      DO 101 J=1,NN
      C(I,J)=0
  101 CONTINUE
  102 CONTINUE
      MAG4=3
C
C     SET KOUNT(I)=0
C
  109 DO 103 I=1,NUM
      KOUNT(I)=0
  108 CONTINUE
C
C     READ ONE TITILE FROM TAPE
C
      NUM1=NUM-1
  110 READ(MAG4)N,(ITITL(I),I=1,N)
      IF(N.EQ.4) GOTO 150
C     COMPARE ITITL WITH KEYBAR
C     INCREMENT KOUNT(I)
C
  112 N1=N-1
      DO 120 I=5,N1,2
      DO 115 J=1,NUM1,2
      IF(ITITL(I).NE.KEYBAR(J)) GOTO 115
      IF(ITITL(I+1).NE.KEYBAR(J+1))GOTO 115
      KOUNT(J)=1
      GOTO 120
  115 CONTINUE
  120 CONTINUE
C
C     INCREMENT C(I,J)
      DO 130 I=1,NUM1,2
      M=I/2+1
      DO 125 J=1,NUM1,2
      IF(KOUNT(J).NE.1) GOTO 125
      L=J/2+1
      IF(KOUNT(I).EQ.1) C(M,L)=C(M,L)+1
  125 CONTINUE
  130 CONTINUE
      GOTO 109
C
C     PRINT C(I,J)
  150 I=1
      WRITE(6,151)((I,J,C(I,J),J=1,NN),I=1,NN)
  151 FORMAT('0',10X,3I4,10X,3I4,10X,3I4,10X,3I4,10X,3I4,10X)
C
C     FIND INVERSE OF MATRIX MAT(M,N)
      CALL FINVER(NN,MAT,INVERS,RHS,XX,BB,RR,PIVOT)
      DO 444 I=1,NA,2
      WRITE(6,200)I,KEYBAR(I),KEYBAR(I+1),KLBR,WTT(I),KRBR
```

```
  200 FORMAT(/5X,I2,1X,2A4,1X,A1,'WT=',F10.5,A1)
  444 CONTINUE
C
C      FIND RESPONSE TO EACH HYPOTHETICAL DOCUMENT
      DO 185 L=1,KAPM
      DO 170 I=1,NUM
      KOUNT(I)=0
  170 CONTINUE
      KK=NOTERM(L)*2-1
      DO 175 I=1,KK,2
      DO 174 J=1,NUM1,2
      IF(KEYWD(I,L).NE.KEYBAR(J)) GOTO 174
      IF(KEYWD(I+1,L).NE.KEYBAR(J+1)) GOTO 174
      KOUNT(J)=1
  174 CONTINUE
  175 CONTINUE
      RESP=0.
      DO 178 I=1,NA,2
      RESP=RESP+WTT(I)*KOUNT(I)
  178 CONTINUE
      WRITE(6,180) RESP
  180 FORMAT(/5X,'RESPONSE TO A HYPOTHETICAL DOCU IS',F16.8,1X,'(EXACT
     U SOLUTION)')
  185 CONTINUE
C
C      FIND RESPONSE TO EACH DOCUMENT IN TAPE
      REWIND MAG4
      KK=1
  300 RESP=0.
  201 DO 202 I=1,NUM
      KOUNT(I)=0
  202 CONTINUE
C
C      READ ONE TITILE FROM TAPE
      READ(MAG4)N,(ITITL(I),I=1,N)
      IF (N.EQ.4) GOTO 250
C
C      COMPARE ITITL WITH KEYBAR, INCREMENT KOUNT(I)
      N1=N-1
      DO 210 I=5,N1,2
      DO 207 J=1,NA,2
      IF(ITITL(I).NE.KEYBAR(J)) GOTO 207
      IF(ITITL(I+1).NE.KEYBAR(J+1)) GOTO 207
      KOUNT(J)=1
  207 CONTINUE
  210 CONTINUE
      DO 215 J=1,NA,2
      RESP=RESP+WTT(J)*KOUNT(J)
  215 CONTINUE
      IF(RESP.LE.0.) GOTO 300
C
C      ARRANGE OUTPUT IN DESCENDING ORDER OF R(N)
      STORE(KK)=RESP
      ITEM(KK,1)=N
      DO 292 I=1,N
```

```
      ITEM(KK,I+1)=ITITL(I)
  292 CONTINUE
      IF (KK.EQ.1) GOTO 285
      LK=KK
  294 IF(STORE(LK).LE.STORE(LK-1)) GOTO 296
      IF(ITEM(LK,1).GT.ITEM(LK-1,1)) GOTO 282
  281 NN=ITEM(LK-1,1)
      GOTO 284
  282 NN=ITEM(LK,1)
  284 DO 283 L=1,NN
      ITEMP=ITEM(LK,L)
      ITEM(LK,L)=ITEM(LK-1,L)
      ITEM(LK-1,L)=ITEMP
  283 CONTINUE
      TEMP=STORE(LK)
      STORE(LK)=STORE(LK-1)
      STORE(LK-1)=TEMP
  295 LK=LK-1
      IF(LK.NE.1) GOTO 294
  296 IF(KK.EQ.100) GOTO 300
  285 KK=KK+1
      GOTO 300
  250 WRITE(6,251)
  251 FORMAT('1'//,12X,'*** TITILES RETRIEVED ***'/)
      DO 288 KK=1,100
      WRITE(6,287) STORE(KK)
  287 FORMAT(/16X,'R(N)=',F12.7)
      M=ITEM(KK,1)
      WRITE(6,289) (ITEM(KK,LL),LL=2,M)
  289 FORMAT(12X,16A4/(28X,12A4))
  288 CONTINUE
  290 WRITE (6,291)
  291 FORMAT(/12X,'*** SEARCH COMPLETED ***')
      REWIND MAG4
      GOTO 10
  556 STOP
      END
```

```
C
C
C
C      THIS ROUTINE COMPUTES THE INVERSE OF A MATRIX BY
C      CALLING A LIBRARY SUBROUTINE CS011A
C
       SUBROUTINE FINVER(N,MAT,INVERS,RHS,XX,BB,RR,PIVOT)
       REAL*4 MAT(N,N),INVERS(N,N),RHS(N),XX(N),BB(N,N),RR(N)
       INTEGER PIVOT(N),C(100,100)
       COMMON WTT(200),IWT(200),C
       DO 601 I=1,N
       DO 600 J=1,N
       MAT(I,J)=C(I,J)
   600 CONTINUE
   601 CONTINUE
       CALL CS011A(MAT,INVERS,RHS,XX,N,BB,RR,PIVOT)
       WRITE(6,602)((I,J,INVERS(I,J),J=1,N),I=1,N)
   602 FORMAT(/5X,'THE FOLLOWING ARE ELEMENTS OF THE INVERSE OF MATRIX
      UC(I,J)'/(4(5X,2I4,F16.8)))
C    FIND EXECT WTS USING EQUATION 10
       NA=2*N
       DO 190 I=1,NA,2
       WTT(I)=0.
   190 CONTINUE
       WRITE(6,199)
   199 FORMAT(/5X,'THE FOLLOWING ARE WTS BY EXACT SOLUTION:')
       DO 195 I=1,NA,2
       M=I/2+1
       DO 194 J=1,NA,2
       L=J/2+1
       WTT(I)=WTT(I)+INVERS(M,L)*IWT(J)
   194 CONTINUE
   195 CONTINUE
       RETURN
       END
```

```
C
C
C
C     *** CROSS-CORRELATED SEARCH ***
C
      DIMENSION ZBAR(100),STORE(100),ITEM(100,53)
      DIMENSION YBAR(100),T(100)
      DIMENSION NOTERM(20),KEYWD(30,12),NUMI(20),KOUNT(200),ITITL(100),
     1KEYBAR(100),C(100,100),IWT(100)
      DATA KBLAN/'    '/,KZZZZ/'ZZZZ'/,KLBR/'(   '/,KRBR/')   '/,
     1KCOM/',   '/
C
C     READ KEYWORDS OF QUESTION
   10 READ(5,21,END=556) KAPM,ITRACE
   21 FORMAT(2I4)
      NUM=0
      DO 35 M=1,KAPM
      I1=1
   24 I2=I1+25
      READ(5,25)(KEYWD(I,M),I=I1,I2)
   25 FORMAT(13(A4,A2))
      I3=I2-1
      DO 28 I=I1,I3,2
      IF(KEYWD(I,M).EQ.KBLAN) GOTO 30
   28 CONTINUE
      I1=I1+26
      GOTO 24
   30 K=I-1
      WRITE (6,333)
  333 FORMAT(/5X,'THE FOLLOWING ARE KEYWORDS FROM A QUESTION')
      WRITE(6,334)(KEYWD(II,M),II=I1,K)
  334 FORMAT (15X,2A4)
      NOTERM(M)=K/2
      WRITE(6,555) NOTERM(M)
  555 FORMAT (5X,'NO. OF TERMS=',I2)
      NUMI(M)=I-1
      NUM=NUM+NUMI(M)
   35 CONTINUE
      IF(NUM.LE.200) GOTO 37
      WRITE(6,36)
   36 FORMAT(5X,' MORE THAN 100 KEYWORDS READ ')
      STOP
   37 IF(ITRACE.NE.1) GOTO 40
      WRITE(6,38) KAPM
   38 FORMAT(5X,' KAPM=',I4)
      WRITE(6,39) NUM
   39 FORMAT(5X,'NUM=',I4/)
C
C     COMBINE KEYWORDS TO FORM KEYBAR(I) IN
C     ALPHABETIC ORDER WITH WEIGHTS IWT(I)
C     FIRST MERGE KEYWORDS
   40 I1=1
   44 DO 50 M=1,KAPM
      I2=I1+NUMI(M)-1
      DO 48 I=I1,I2
```

```
      KEYBAR(I)=KEYWD(I-I1+1,M)
   48 CONTINUE
      I1=I2+1
   50 CONTINUE
      NUM=I1-1
C     SET IWT(I)=1 FOR I=1,NUM-1 IN STEPS OF 2
      J=NUM-1
      DO 55 I=1,J,2
      IWT(I)=1
   55 CONTINUE
C
C     BUBBLE SORT KEYBAR(I)
      DO 78 K=1,J,2
      I=NUM-1
      LAST=0
   62 IF(KEYBAR(I) - KEYBAR(I-2)) 66,64,76
   64 IF(KEYBAR(I+1)-KEYBAR(I-1)) 66,70,76
   66 IT=KEYBAR(I)
      KEYBAR(I)=KEYBAR(I-2)
      KEYBAR(I-2)=IT
      IT=KEYBAR(I+1)
      KEYBAR(I+1)=KEYBAR(I-1)
      KEYBAR(I-1)=IT
      IT=IWT(I)
      IWT(I)=IWT(I-2)
      IWT(I-2)=IT
      LAST=1
      GOTO 76
   70 IWT(I-2)=IWT(I-2)+IWT(I)
      IWT(I)=0
      KEYBAR(I)=KZZZZ
      KEYBAR(I+1)=KBLAN
      LAST=1
   76 I=I-2
      IF(I.GT.K) GOTO 62
      IF(LAST.EQ.0) GOTO 80
   78 CONTINUE
C
C     ELIMINATE DUPLICATE KEYWORDS
      DO 79 I=1,J,2
      IF (KEYBAR(I).EQ.KZZZZ) GOTO 77
   79 CONTINUE
      GOTO 80
   77 NUM=I-1
   80 IF(ITRACE.NE.1) GOTO 100
      WRITE(6,39) NUM
      NA=NUM-1
      WRITE(6,90) (I,KEYBAR(I),KEYBAR(I+1),KLBR,IWT(I),KRBR,KCOM ,
     AI=1,NA,2)
   90 FORMAT (5(5X,I3,1X,2A4,1X,A1,I2,2A1)/)
C
C     SET COUNTER C(I,J)=0
      NK=NUM/2
      NUM1=NUM-1
      NN=NK**2
```

```
  100 DO 102 I=1,NN
      YBAR(I)=0.
      DO 101 J=1,NN
      C(I,J)=0.
  101 CONTINUE
  102 CONTINUE
C
C     FIND YBAR IN TERMS OF KEYBAR
      DO 760 LM=1,KAPM
      DO 705 I=1,NUM,2
      KOUNT(I)=0
  705 CONTINUE
      KK=NOTERM(LM)*2-1
      DO 715 J=1,KK,2
      DO 714 M=1,NUM1,2
      IF(KEYWD(J,LM).NE.KEYBAR(M)) GOTO 714
      IF(KEYWD(J+1,LM).NE.KEYBAR(M+1)) GOTO 714
      KOUNT(M)=1
  714 CONTINUE
  715 CONTINUE
      DO 160 L=1,NK
      LL=L*2-1
      IF(KOUNT(LL).EQ.0) GOTO 160
      DO 162 M=1,NK
      MM=M*2-1
      IF(KOUNT(MM).EQ.0) GOTO 162
      K=(L-1)*NK+M
      IF(L.EQ.M) GOTO 165
      YBAR(K)=YBAR(K)+KOUNT(LL)*KOUNT(MM)/2.
      GOTO 742
  165 YBAR(K)=YBAR(K)+KOUNT(LL)
  742 WRITE(6,750)K,L,M,YBAR(K)
  750 FORMAT(5X,'K=',I3,2X,'I=',I3,2X,'J=',I3,2X,F10.5)
  162 CONTINUE
  160 CONTINUE
  760 CONTINUE
      WRITE(6,765) (K,YBAR(K),K=1,NN)
  765 FORMAT('0',5(8X,I4,2X,F7.3))
C
C     SET COUNTER KOUNT(I)=0
      MAG4=3
  109 DO 108 I=1,NUM
      KOUNT(I)=0
  108 CONTINUE
      DO 168 I=1,NN
      ZBAR(I)=0.
  168 CONTINUE
C
C     READ ONE TITILE FROM TAPE
  110 READ(MAG4)N,(ITITL(I),I=1,N)
      IF(N.EQ.4) GOTO 150
C
C     COMPARE ITITL WITH KEYBAR
C     INCREMENT KOUNT(I)
      IFLAG=0
```

```
  112 N1=N-1
      DO 120 I=5,N1,2
      DO 115 J=1,NUM1,2
      IF(ITITL(I).NE.KEYBAR(J)) GOTO 115
      IF(ITITL(I+1).NE.KEYBAR(J+1))GOTO 115
      KOUNT(J)=1
      IFLAG=IFLAG+1
      GOTO 120
  115 CONTINUE
  120 CONTINUE
      IF(IFLAG.LE.0) GOTO 110
C
C     FIND Z(K) IN TERMS OF Z(I,J)
      DO 172 I=1,NK
      II=I*2-1
      IF(KOUNT(II).EQ.0) GOTO 172
      DO 175 J=1,NK
      JJ=J*2-1
      IF(KOUNT(JJ).EQ.0) GOTO 175
      K=(I-1)*NK+J
      IF(I.EQ.J) GOTO 178
      ZBAR(K)=KOUNT(II)*KOUNT(JJ)/2.
      GOTO 175
  178 ZBAR(K)=KOUNT(II)
  175 CONTINUE
  172 CONTINUE
      DO 180 M=1,NN
      IF(ABS(ZBAR(M)).LE.0.0001) GOTO 180
      DO 185 N=1,NN
      IF(ABS(ZBAR(N)).LE.0.0001) GOTO 185
      C(M,N)=C(M,N)+ZBAR(M)*ZBAR(N)
  185 CONTINUE
  180 CONTINUE
      GOTO 109
  150 SQM=0.
      DO 230 I=1,NN
      IF(ABS(C(I,I)).LE.0.0001) GOTO 230
      AA=YBAR(I)/C(I,I)
      AB=0.
      DO 220 J=1,NN
      IF(J.EQ.I) GOTO 220
      IF(ABS(C(J,J)).LE.0.0001) GOTO 220
      AB=AB+C(I,J)*YBAR(J)/C(I,I)*C(J,J)
  220 CONTINUE
      T(I)=AA-AB
      WRITE(6,225) I,T(I)
  225 FORMAT(/5X,'WT-ELEMENT(',I2,')=',F16.5)
C
C     FIND MEAN SQUARE OF WHOLE COLLECTION
      SQM=SQM+T(I)*YBAR(I)
  230 CONTINUE
      WRITE(6,190) SQM
  190 FORMAT(/5X,'MEAN SQUARE VALUE OF COLLECTION :',F16.5)
      REWIND MAG4
      KK=1
```

```
  336 DO 338 I=1,NUM,2
      KOUNT(I)=0
  338 CONTINUE
      DO 339 J=1,NN
      ZBAR(J)=0.
  339 CONTINUE
  341 READ(MAG4)N,(ITITL(I),I=1,N)
      IF(N.EQ.4) GOTO 400
  887 IFLAG=0
      DO 340 I=5,N,2
      DO 342 J=1,NUM1,2
      IF(ITITL(I).NE.KEYBAR(J)) GOTO 342
      IF(ITITL(I+1).NE.KEYBAR(J+1)) GOTO 342
      KOUNT(J)=1
      IFLAG=IFLAG+1
      GOTO 340
  342 CONTINUE
  340 CONTINUE
      IF(IFLAG.LE.0) GOTO 341
C
C     FIND ZBAR(R) IN TERMS OF Z(I,J)
      DO 350 I=1,NK
      II=I*2-1
      IF(KOUNT(II).EQ.0) GOTO 350
      DO 355 J=1,NK
      JJ=J*2-1
      IF(KOUNT(JJ).EQ.0) GOTO 355
      K=(I-1)*NK+J
      IF(I.EQ.J) GOTO 353
      ZBAR(K)=KOUNT(II)*KOUNT(JJ)/2.
      GOTO 355
  353 ZBAR(K)=KOUNT(II)
  355 CONTINUE
  350 CONTINUE
      RESP=0.
      DO 378 I=1,NN
      IF(ABS(ZBAR(I)).LE.0.0001) GOTO 378
      RESP=RESP+T(I)*ZBAR(I)
  378 CONTINUE
      IF(ABS(RESP).LE.0.001)GOTO 336
      TS=RESP/SQM
C
C     ARRANGE OUTPUT IN DESCENDING ORDER OF R(N)
      STORE(KK)=TS
      ITEM(KK,1)=N
      DO 380 I=1,N
      ITEM(KK,I+1)=ITITL(I)
  380 CONTINUE
      IF(KK.EQ.1) GOTO 385
      LK=KK
  394 IF(STORE(LK).LE.STORE(LK-1)) GOTO 396
      IF(ITEM(LK,1) .GT.ITEM(LK-1,1)) GOTO 382
      MM=ITEM(LK-1,1)
      GOTO 384
  382 MM=ITEM(LK,1)
```

```
384 DO 383 L=1,MM
    ITEMP=ITEM (LK,L)
    ITEM(LK,L)=ITEM(LK-1,L)
    ITEM(LK-1,L)=ITEMP
383 CONTINUE
    TEMP=STORE(LK)
    STORE(LK)=STORE(LK-1)
    STORE(LK-1)=TEMP
    LK=LK-1
    IF(LK.NE.1) GOTO 394
396 IF(KK.EQ.100) GOTO 386
385 KK=KK+1
    GOTO 336
386 KK=100
    GOTO 336
400 DO 388 KK=1,100
    M=ITEM(KK,1)
    WRITE(6,387) STORE(KK)
387 FORMAT(/5X,'RESPONSE RATIO RESP/SQM=',F15.7)
    WRITE(6,389)(ITEM(KK,LL),LL=2,M)
389 FORMAT(1X,33A4/)
388 CONTINUE
390 WRITE(6,391)
391 FORMAT(5X,'*** SEARCH COMPLETED ***')
    REWIND MAG4
    GOTO 10
556 STOP
    END
```

```
C
C
C
C     *** LEAST SQUARE SEARCH ***
C
      REAL MAT(20,20),INVERS(20,20),RHS(20),BB(20,20),RR(20),XX(20)
      INTEGER PIVOT(20)
      COMMON V(100),WTT(200),U(50,50)
      DIMENSION F(15),NUMI(15),KOUNT(15,50),NOTERM(20),
     1KOUNT1(100),ITITL(200),STORE(100),ITEM(100,50),KEYWD(20,40),
     2KEYBAR(200)
      DATA KBLAN/'    '/,KZZZZ/'ZZZZ'/,KCOM/',   '/
   10 READ(5,2,END=555) KAPM
    2 FORMAT(I4)
      NUM=0
      DO 25 M=1,KAPM
C
C     READ IN THE TITILES OF QUESTION
C
      READ(5,3) F(M)
    3 FORMAT(F4.0)
      I1=1
    4 I2=I1+25
      READ(5,8)(KEYWD(I,M),I=I1,I2)
    8 FORMAT(13(A4,A2))
      I3=I2-1
      DO 12 I=I1,I3,2
      IF(KEYWD(I,M).EQ.KBLAN) GOTO 20
   12 CONTINUE
      I1=I1+26
      GOTO 4
   20 K=I-1
      WRITE(6,21)
   21 FORMAT(15X,'THE FOLLOWING ARE KEYWORDS FROM A DOCU')
      WRITE(6,22)(KEYWD(I,M),I=I1,K)
   22 FORMAT(15X,2A4)
      WRITE(6,23) F(M)
   23 FORMAT(15X,'F(M)=',F5.2/)
      NUMI(M)=K
      NUM=NUM+NUMI(M)
      NOTERM(M)=K/2
   25 CONTINUE
      WRITE(6,28) NUM
   28 FORMAT(5X,'NUM=',I4/)
C
C     COMBINE KEYWORDS TO FORM KEYBAR IN ALPHABETIC
C     ORDER,AND MERGE THE KEYWORDS
C
      I1=1
      DO 35 M=1,KAPM
      I2=I1+NUMI(M)-1
      DO 30 I=I1,I2
      KEYBAR(I)=KEYWD(I-I1+1,M)
   30 CONTINUE
      I1=I2+1
```

```
   35 CONTINUE
      NUM=I1-1
      J=NUM-1
C
C     BUBBLE SORT KEYBAR(I)
      DO 50 K=1,J,2
      I=NUM-1
      LAST=0
   40 IF(KEYBAR(I)-KEYBAR(I-2)) 42,44,46
   44 IF(KEYBAR(I+1)-KEYBAR(I-1)) 42,48,46
   42 IT=KEYBAR(I)
      KEYBAR(I)=KEYBAR(I-2)
      KEYBAR(I-2)=IT
      IT=KEYBAR(I+1)
      KEYBAR(I+1)=KEYBAR(I-1)
      KEYBAR(I-1)=IT
      LAST=1
      GOTO 46
   48 KEYBAR(I)=KZZZZ
      KEYBAR(I+1)=KBLAN
      LAST=1
   46 I=I-2
      IF(I.GT.K) GOTO 40
      IF(LAST.EQ.0) GOTO 52
   50 CONTINUE
      DO 51 I=1,J,2
      IF(KEYBAR(I).EQ.KZZZZ) GOTO 53
   51 CONTINUE
      GOTO 52
   53 NUM=I-1
   52 WRITE(6,28) NUM
      NA=NUM-1
      WRITE(6,54)((I,KEYBAR(I),KEYBAR(I+1),KCOM),I=1,NA,2)
   54 FORMAT(6(5X,I2,1X,2A4,1X,A1)/)
C
C     SET U(I,J)=0, U IS THE CROSS CORRELATION,
C     SET V(I)=0, V IS MEAN ESTIMATED WT OF ITH WORD
      DO 60 I=1,NUM,2
      DO 58 J=1,NUM,2
      U(I,J)=0.
   58 CONTINUE
      V(I)=0.
   60 CONTINUE
      DO 63 N=1,KAPM
      DO 62 M=1,NUM,2
      KOUNT(N,M)=0
   62 CONTINUE
   63 CONTINUE
      DO 70 L=1,KAPM
      KK=NOTERM(L)*2-1
      DO 67 I=1,KK,2
      DO 68 J=1,NA,2
      IF(KEYWD(I,L).NE.KEYBAR(J)) GOTO 68
      IF(KEYWD(I+1,L).NE.KEYBAR(J+1)) GOTO 68
      KOUNT(L,J)=1
```

```
   68 CONTINUE
   67 CONTINUE
   70 CONTINUE
      DO 75 I=1,NA,2
      DO 73 L=1,KAPM
         V(I)=    V(I)+F(L)*KOUNT(L,I)
   73 CONTINUE
   75 CONTINUE
      DO 80 I=1,NA,2
      DO 78 J=1,NA,2
      DO 76 L=1,KAPM
      U(I,J)=U(I,J)+KOUNT(L,I)*KOUNT(L,J)
   76 CONTINUE
   78 CONTINUE
   80 CONTINUE
      WRITE(6,90) (    V(I),I=1,NA,2)
   90 FORMAT(4X,'THE FOLLOWING ARE    V(I)' / (4X,5F16.8))
      WRITE(6,91)((I,J,U(I,J),J=1,NA,2),I=1,NA,2)
   91 FORMAT(/5X,'THE FOLLOWING ARE U(I,J)'/(4(5X,2I4,F16.8)))
C
C     FIND INVERSE OF MATRIX MAT(M,N)
C
      N=NUM/2
      CALL FINVER(  NA,N,MAT,INVERS,RHS,XX,BB,RR,PIVOT)
      DO 444 I=1,NA,2
      WRITE(6,200)I,KEYBAR(I),KEYBAR(I+1),KLBR,WTT(I),KRBR
  200 FORMAT(/5X,I2,1X,2A4,1X,A1,'WT=',F10.5,A1)
  444 CONTINUE
C
C     FIND RESPONSE TO EACH DOCUMENT
C
      MAG4=3
      KK=1
  300 RESP=0.
  201 DO 202 I=1,NUM
      KOUNT1(I)=0
  202 CONTINUE
C
C     READ ONE TITILE FROM TAPE
      READ(MAG4) N, (ITITL(I),I=1,N)
      IF (N.EQ.4) GOTO 250
C
C     COMPARE ITITL WITH KEYBAR, INCREMENT KOUNT(I)
      N1=N-1
      DO 210 I=5,N1,2
      NUM1=NUM-1
      DO 207 J=1,NUM1,2
      IF(ITITL(I).NE.KEYBAR(J)) GOTO 207
      IF(ITITL(I+1).NE.KEYBAR(J+1)) GOTO 207
      KOUNT1(J)=1
      GOTO 210
  207 CONTINUE
  210 CONTINUE
      DO 215 J=1,NUM1,2
      RESP=RESP+WTT(J)*KOUNT1(J)
```

```
  215 CONTINUE
      IF(RESP.LT.0.) GOTO 300
C
C     ARRANGE OUTPUT IN DESCENDING ORDER OF R(N)
      STORE(KK)=RESP
      ITEM(KK,1)=N
      DO 292 I=1,N
      ITEM(KK,I+1)=ITITL(I)
  292 CONTINUE
      IF (KK.EQ.1) GOTO 285
      LK=KK
 294  IF(STORE(LK)  .LE.STORE(LK-1)) GOTO 296
      IF(ITEM(LK,1)-ITEM(LK-1,1)) 281,282,282
  281 NN=ITEM(LK-1,1)
      GOTO 284
  282 NN=ITEM(LK,1)
  284 DO 283 L=1,NN
      ITEMP=ITEM(LK,L)
      ITEM(LK,L)=ITEM(LK-1,L)
      ITEM(LK-1,L)=ITEMP
  283 CONTINUE
      TEMP=STORE(LK)
      STORE(LK)=STORE(LK-1)
      STORE(LK-1)=TEMP
  295 LK=LK-1
      IF(LK.NE.1) GOTO 294
  296 IF(KK.EQ.100) GOTO 286
  285 KK=KK+1
      GOTO 300
  286 KK=100
      GOTO 300
  250 DO 288 KK=1,100
      M=ITEM(KK,1)
      WRITE(6,287) STORE(KK)
  287 FORMAT(/5X,' RESP=',F15.7)
      WRITE(6,289)(ITEM(KK,LL),LL=2,M)
  289 FORMAT(1X,33A4/)
  288 CONTINUE
  290 WRITE (6,291)
  291 FORMAT(5X,'*** SEARCH COMPLETED***')
      REWIND MAG4
      GOTO 10
  555 STOP
      END
```

```
C
C     *** 2-PARAMETER SEARCH ***
C
      DIMENSION KEYWD(40,2),KT1(40),KT2(40),ITITL(100)
      REAL MAT(20,20),INVERS(20,20),RHS(20),BB(20,20),RR(20),XX(20)
      INTEGER PIVOT(20),C(20,20),ITEM(100,56),KBLAN/'     '/
      REAL V1(20),V2(20),STORE(100)
C
C     READ KEYWORDS IN THE 2 VECTORS
    1 DO 20 M=1,2
      I1=1
    2 I2=I1+25
      READ(5,7,END=555)(KEYWD(I,M),I=I1,I2)
    7 FORMAT(13(A4,A2))
        DO 8 I=I1,I2,2
      IF(KEYWD(I,M).EQ.KBLAN) GOTO 10
    8 CONTINUE
      I1=I1+26
      GOTO 2
C
C     K,NUM INDICATING THE LAST PART OF LAST WORD
   10 K=I-1
      WRITE(6,5)
    5 FORMAT(/5X,'KEYWORDS OF AN INPUT VECTOR'/)
      WRITE(6,12) (KEYWD(I,M),I=I1,K)
   12 FORMAT(/5X,2A4)
      NUM=K
      NTERM=K/2
      WRITE(6,15) NTERM
   15 FORMAT(/5X,'NO. OF TERMS =',I4/)
   20 CONTINUE
      DO 444 I=1,20
      DO 443 J=1,20
      C(I,J)=0
  443 CONTINUE
  444 CONTINUE
      MAG=3
      GOTO 45
   99 WRITE(6,199)
  199 FORMAT(5X ,'*** ERROR ***')
C
C     INITIALISE KT1,KT2
   45 DO 48 I=1,NUM,2
      KT1(I)=0
      KT2(I)=0
   48 CONTINUE
C
C     READ ONE TITILE FROM TAPE
      NA=NUM-1
      READ(MAG,ERR=99) N,(ITITL(I),I=1,N)
      IF(N.EQ.4) GOTO 500
C
C     COMPARE ITITL WITH KEYWD(I,M)
      N1=N-1
      DO 70 K=5,N1,2
```

```
      DO 68 I=1,NA,2
      IF(ITITL(K).NE.KEYWD(I,1)) GOTO 53
      IF(ITITL(K+1).NE.KEYWD(I+1,1)) GOTO 53
      KT1(I)=1
   53 IF(ITITL(K).NE.KEYWD(I,2)) GOTO 68
      IF(ITITL(K+1).NE.KEYWD(I+1,2)) GOTO 68
      KT2(I)=1
   68 CONTINUE
   70 CONTINUE
C
C     INCREMENT C(I,J)
      DO 80 I=1,NUM,2
      L=I/2+1
      IF(KT1(I).NE.1) GOTO 80
      DO 75 J=1,NUM,2
      M=J/2+1
      IF(KT2(J).NE.1) GOTO 75
      C(L,M)=C(L,M)+1
   75 CONTINUE
   80 CONTINUE
      GOTO 45
  500 WRITE(6,85)((I,J,C(I,J),J=1,NTERM),I=1,NTERM)
   85 FORMAT(/(10X,3I4))
C
C     FIND INVERSE OF C, MAT IS C IN REAL MODE
      CALL FINVA(NTERM,MAT,INVERS,RHS,XX,BB,RR,PIVOT,C,V1,V2)
      SUMV1=0.
      DO 106 I=1,NTERM
      SUMV1=SUMV1+V1(I)
  106 CONTINUE
      WRITE(6,109) SUMV1
  109 FORMAT(/5X,'SUMV1=',F16.6)
C
C     FIND RESPONSE TO A GENENAL DOCUMENT
C     READ A TITIL AND DETERMINE ITS KEYWORDS
C     WITH RESPECT TO P1 AND P2
      REWIND MAG
      NC=0
      KK=1
  100 RESP=0.
      A1=0.
      A2=0.
      DO 107 I=1,NUM,2
      KT1(I)=0
      KT2(I)=0
  107 CONTINUE
      READ(MAG) N,(ITITL(I),I=1,N)
      IF(N.EQ.4) GOTO 600
      N1=N-1
      DO 115 K=5,N1,2
      DO 110 I=1,NA,2
      IF(ITITL(K).NE.KEYWD(I,1)) GOTO 108
      IF(ITITL(K+1).NE.KEYWD(I+1,1)) GOTO 108
      KT1(I)=1
  108 IF(ITITL(K).NE.KEYWD(I,2)) GOTO 110
```

```
      IF(ITITL(K+1).NE.KEYWD(I+1,2)) GOTO 110
      KT2(I)=1
  110 CONTINUE
  115 CONTINUE
      DO 120 I=1,NA,2
      M=I/2+1
      A1=A1+V1(M)*KT1(I)
      A2=A2+Y2(M)*KT2(I)
  120 CONTINUE
      IF(ABS(A1).LE.0.00001.OR.ABS(A2).LE.0.00001) GOTO 100
      WRITE(6,121) A1,A2
  121 FORMAT(/5X,'A1=',F16.6,10X,'A2=',F10.6)
      NO=NO+1
      RESP=A1*A2/SUMV1
C
C     ARRANGE OUTPUT IN DESCENDING ORDER OF R(N)
      STORE(KK)=RESP
      ITEM(KK,1)=N
      DO 130 I=1,N
      ITEM(KK,I+1)=ITITL(I)
  130 CONTINUE
      IF(KK.EQ.1) GOTO 152
      LK=KK
  132 IF(ABS(STORE(LK)).LE.ABS(STORE(LK-1)))GOTO 150
      IF(ITEM(LK,1).GT.ITEM(LK-1,1)) GOTO 140
  137 NN=ITEM(LK-1,1)
      GOTO 142
  140 NN=ITEM(LK,1)
  142 DO 145 L=1,NN
      ITEMP=ITEM(LK,L)
      ITEM(LK,L)=ITEM(LK-1,L)
      ITEM(LK-1,L)=ITEMP
  145 CONTINUE
      TEMP=STORE(LK)
      STORE(LK)=STORE(LK-1)
      STORE(LK-1)=TEMP
      LK=LK-1
      IF(LK.NE.1) GOTO 132
  150 IF(KK.EQ.100) GOTO 100
  152 KK=KK+1
      GOTO 100
  600 IF(NO.GT.100) NO=100
      DO 169 KK=1,NO
      WRITE(6,162) STORE(KK)
  162 FORMAT(/5X,'RESP=',F15.7)
      M=ITEM(KK,1)
      WRITE(6,165)(ITEM(KK,L),L=2,M)
  165 FORMAT(1X,33A4//)
  169 CONTINUE
      WRITE(6,175)
  175 FORMAT(/5X,'*** SEARCH COMPLETED ***')
      REWIND MAG
      GOTO 1
  555 STOP
      END
```

```
C
C
C
C     THIS ROUTINE INVERSE A MATRIX AND COMPUTE
C     THE WEIGHTS BY CALLING THE LIBRARY SUBROUTINE CS011A
C
      SUBROUTINE FINVA(N,MAT,INVERS,RHS,XX,BB,RR,PIVOT,C,V1,V2)
      REAL V1(20),V2(20)
      REAL*4 MAT(N,N),INVERS(N,N),RHS(N),XX(N),BB(N,N),RR(N)
      INTEGER PIVOT(N),C(20,20)
      DO 601 I=1,N
      DO 600 J=1,N
      MAT(I,J)=C(I,J)
  600 CONTINUE
  601 CONTINUE
      CALL CS011A(MAT,INVERS,RHS,XX,N,BB,RR,PIVOT)
      WRITE(6,602)((I,J,INVERS(I,J),J=1,N),I=1,N)
  602 FORMAT(/5X,'THE FOLLOWING ARE ELEMENTS OF THE INVERSE OF MATRIX
     UC(I,J)'/(4(5X,2I4,F16.8)))
      DO 101 I=1,N
      V1(I)=0.
      V2(I)=0.
  101 CONTINUE
      DO 105 I=1,N
      DO 103 J=1,N
      V1(I)=V1(I)+INVERS(J,I)
      V2(I)=V2(I)+INVERS(I,J)
  103 CONTINUE
      WRITE(6,104) V1(I),V2(I)
  104 FORMAT(/5X,'V1(I)=',F16.6,10X,'V2(I)=',F16.6)
  105 CONTINUE
      RETURN
      END
```

APPENDIX B

In response to the question concerning machine, engine, and fan, noise the matched search produced a list of 100 titles arranged in descending order of relevance as indicated by R(N). The following pages contain a portion of that list.

```
*** TITILES RETRIEVED ***


     R(N)=    0.6666667
ACC 55   027   217BERAN   KAMPE    ALLEN    NOISE    CENTR    FANS


     R(N)=    0.6666667
ACO 56   028   681NIEKE   MEASU    NOISE    DUCTE    FANS

     R(N)=    0.6666667
ACO 58   030  0714HOWES   REAL     NOISE    ORIGI    POWER    SPECT
                   DUCTE   CENTR    FANS

     R(N)=    0.3562517
ACO 66   C39   907BISHO   REDUC    AIRCR    NOISE    MEASU    SEVER
                   SCHOO   MOTEL    RESID    ROOMS

     R(N)=    0.2416164
ACO 56   028   427PIETR   JET      NOISE    PROBL    AIRCR    CARRI
                   ISLAN

     R(N)=    0.2416164
ACO 56   C28   434PIETR   NOISE    MEASU    SOME     JET      AIRCR


     R(N)=    0.2416164
ACO 57   029  0854KIRCH   GREEN    NONLI    RESPO    AIRCR    PANEL
                   ACOUS   NOISE

     R(N)=    0.2416164
ACU 63    13   324ROBIN   BOWSH    COPEL    JUDGI    NOISE    AIRCR
                   FLIGH

     R(N)=    0.2416164
SOV 65    11   170MELNI   NOISE    GENER    GROUN    DURIN    TAKEO
                   LANDI   TU124    PASSE    AIRCR

     R(N)=    0.2416164
ACU 65    16   251KOPPE   MATSC    MULLE    PROCE    DESCR    ASSES
                   AIRCR   NOISE    VICIN    AIRPO

     R(N)=    0.2416164
ACO 66   039   138WILLI   KRYTE    MASKI    SPEEC    AIRCR    NOISE


     R(N)=    0.2416164
ACO 66   040   108BISHO   JUDGM    RELAT    ABSOL    ACCEP    AIRCR
                   NOISE

     R(N)=    0.2369385
ACO 57   029  1289WARD    HEARI    NAVAL    AIRCR    MAINT    PERSO


     R(N)=    0.2369385
ACO 59   031  1415KRYTE   SCALI    HUMAN    REACT    SOUND    AIRCR


     R(N)=    0.2369385
ACU 66    17   245BOWSH   JOHNS    ROBIN    FURTH    EXPER    JUDGI
```

```
ACC 61   033 1558LOEB     FLETC     CONTR     THRES     SHIFT     REDUC
                  TEMPO     THRES     SHIFT     INDIC     ACOUS     REFLE
                  ACTIO

         R(N)=    0.1367636
ACO 58   030 0533JILLI    FIALA     METHO     GENER     HIGH      INTEN
                  SOUND     LOUDS     ENVIR     TESTI     ELECT     COMPO
                  SUBJE     JET       MISSI     ENGIN     NOISE

         R(N)=    0.1367636
ACO 59   031 0420MAGLI    SHIEL     FLAP      TYPE      JET       ENGIN
                  NOISE     SUPPR

         R(N)=    0.1367636
ACO 59   031 1013MAYES    SOME      NEAR      FAR       FIELD     NOISE
                  MEASU     ROCKE     ENGIN     OPERA     DIFFE     NOZZL
                  PRESS     RATIO

         R(N)=    0.1367636
SOV 65    10  276MELNI    EFFEC     FLIGH     VELOC     ENGIN     NOISE
                  TU124     TRANS

         R(N)=    0.1320857
ACO 55   027  449WATTE    LABAT     BERAN     ACOUS     BEHAV     SOME
                  ENGIN     TEST      CELL      STRUC

         R(N)=    0.1320857
ACC 56   028  688WATER    COOK      BEREN     MEASU     SOUND     ABSOR
                  JET       ENGIN     TEST      CELLS

         R(N)=    0.1320857
SOV 56   002  335ROZEN    HUTER     BOLT      SONIC     TECHN     USE
                  SOUND     ULTRA     ENGIN     SCIEN

         R(N)=    0.1320857
SOV 56   002  337CRAWF    ULTRA     ENGIN

         R(N)=    0.1320857
SOV 57   003   91KRASI    BERGM     ULTRA     APPLI     SCIEN     ENGIN


         R(N)=    0.1320857
ACO 58   030 0378HUETE    SONIC     ENGIN

         R(N)=    0.1320857
ACO 59   031 0065KAMPS    STATI     EVALU     NEAR      FIELD     SOUND
                  PRESS     GENER     EXHAU     HIGH      PERFO     JET
                  ENGIN

         R(N)=    0.1320857
ACO 64   036 2027TEDRI    ACOUS     MEASU     STSTI     TESTS     CLUDT
                  SINGL     NOZZL     ROCKE     ENGIN

         R(N)=    0.1320857
ACO 65   037 0444HARRI    HEARI     LOSS      TREND     CURVE     DAMAG
                  RISK      CRITE     DIESE     ENGIN     PERSO

         R(N)=    0.1320857
SOV 65    10  423CHUGU    CONFE     APPLI     ULTRA     INTEN     CHEMI
                  ENGIN     PROCE

         R(N)=    0.1320857
ACO 67    42 1244WANG     ACOUS     ANALY     MULTI     CYLIN     ENGIN
```

APPENDIX C

From equation (2.6), differentiating with respect to w_i yields

$$\frac{2\ x_i(\sum_i w_i\ \bar{x}_i)[\sum_i \sum_j c_{ij}\ w_i\ w_j] - (\sum_i w_i\ \bar{x}_i)^2[\sum_i c_{ij} w_i + \sum_j c_{ij} w_j]}{(\sum_i \sum_j c_{ij}\ w_i\ w_j)^2} = 0.$$

(A.1)

Since $c_{ij} = c_{ji}$ and the summation over i is identical to the summation over j, it follows that equation (A.1) may be simplified to

$$2\ x_i(\sum_i w_i\ \bar{x}_i)[\sum_i \sum_j c_{ij}\ w_i\ w_j] - 2(\sum w_i\ \bar{x}_i)^2 \sum_i c_{ij} w_i = 0.$$

(A.2)

Defining λ to be the following:

$$\lambda = \frac{(\sum_i w_i\ \bar{x}_i)^2}{\sum_i \sum_j c_{ij}\ w_i\ w_j}$$

and substituting into equation (A.2), equation (2.7) results.

APPENDIX D

$$\text{Since } C = D + N \tag{D.1}$$

$$\begin{aligned} C^{-1} &= (D + N)^{-1} \\ &= [D(I + D^{-1} N)]^{-1} \\ &= (I + D^{-1} N)^{-1} D^{-1} \quad . \end{aligned} \tag{D.2}$$

Substituting equation (D.2) into equation (2.13), equation (2.19) is obtained which appears as follows:

$$\begin{aligned} r &= \underline{\bar{X}}_T \; C^{-1} \; \underline{Z} \\ &= \underline{\bar{X}}_T \; (I + D^{-1} N)^{-1} D^{-1} \; \underline{Z} \; . \end{aligned} \tag{D.3}$$

B29956